# A Decision Support Framework for Machine Learning Applications

Anli du Preez
Department of Industrial Engineering
University of Stellenbosch

Supervisor:
Prof. James Bekker

Thesis presented in fulfilment of the requirements for the degree of Master of Engineering (Industrial Engineering) in the Faculty of Engineering at Stellenbosch University

*M.Eng Industrial*

December 2020

## Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

## Acknowledgements

"I can do all things through Christ who strengthens me"- Philippians 4:13

I would like to express my heartfelt and sincere gratitude to the following people for their contribution towards this thesis:

- Professor James Bekker, my study leader, for your guidance and sharing your life knowledge, wisdom, time and sense of humour. Thank you for taking me under your wing and giving me the opportunity to pursue my master's degree under your supervision. Thank you for your kindness, care and focus to provide your best.
- My loving parents, Alba and Andries du Preez. The completion of thesis would not have been realised without the support, understanding and love of my family. Thank you for always believing in me, even when I struggled to do so myself.
- The friends who have greatly supported me during the completion of my thesis, Lourens Ferreira, Damian Hennessy and Suané Lourens.
- Professor Francois Smit, Moira Thesner and Anne Erikson, for proofreading my thesis document and making helpful suggestions.

# Abstract

Data is currently one of the most critical and influential emerging technologies. Organisations and employers around the globe strive to investigate and exploit the exponential data growth to discover hidden insights in an effort to create value. Value creation from data is made possible through data analytics (DA) and machine learning (ML). The true potential of data is yet to be exploited since, currently, about 1% of generated data is ever actually analysed for value creation. There is a data gap. Data is available and easy to capture; however, the information therein remains untapped yet ready for digital explorers to discover the hidden value in the data. One main factor contributing to this gap is the lack of expert knowledge in the field of DA and ML.

In a survey of 437 companies, 76% indicated an interest to invest in DA and ML technologies over the years of 2015 to 2017. However, in a survey of 400 companies, 4% indicated that they have the right strategic intent, skilled people, resources and data to gain meaningful insights from their data and to act on them. Small, medium and micro enterprises (SMMEs) lack the availability of DA and ML skills in their existing workforce, have limited infrastructure to realise ML and have limited funding to employ ML tools and expertise. They need proper guidance as to how to employ ML in a low-cost, feasible and sustainable way.

This study focused on addressing this data gap by providing a decision support framework for ML algorithms. The goal of this study was therefore to develop and validate a *decision support framework* which considers both the *data characteristics* and the *application type* to enable SMMEs to choose the appropriate ML algorithm for their unique data and application purpose. This study aimed to develop the framework for a semi-skilled analyst, with mathematics, statistics and programming education, who is familiar

with the process of programming, yet has not specialised in the variety of ML algorithms which are available.

This research project followed the Soft Systems Methodology and utilised Jabareen's framework development methodology. Various literature studies were performed on data, DA, application purposes, ML and the process of applying ML. The CRoss-Industry Standard Process for Data Mining (CRISP-DM) was followed to design and implement the experiments. The results were evaluated and summarised to create the decision support framework. The framework was validated by consulting subject matter experts (SMEs) and possible end-users (PEUs).

# Opsomming

Data is tans een van die mees kritieke en invloedrykste ontluikende tegnologieë. In 'n poging om besigheidswaarde te skep, streef organisasies en werkgewers regoor die wêreld daarna om die eksponensiële groei van data te ondersoek en te benut om verborge inligting en insigte te ontdek. Waardeskepping vanuit data word deur data-analise (DA) en masjienleer (ML) moontlik gemaak. Die werklike potensiaal van data moet nog ontgin word, aangesien slegs ongeveer 1% van die gegenereerde data tans vir die ontginning van besigheidswaarde ontleed word. Daar is 'n datagaping. Data is geredelik beskikbaar en maklik om vas te vang, maar die inligting daarin bly onbenut, maar gereed vir digitale ontdekkingsreisigers om die verborge waarde in die data te ontdek. 'n Groot faktor wat tot hierdie gaping bydra, is die gebrek aan kundige kennis op die gebied van DA en ML.

In 'n opname onder 437 maatskappye het 76% 'n belangstelling in die belegging in DA- en ML-tegnologieë in die jare 2015 tot 2017 aangedui. In 'n peiling onder 400 ondernemings het 4 % egter aangedui dat hulle die regte strategiese ingesteldheid, arbeidsmag, hulpbronne en data het om betekenisvolle insigte uit hul data te ontgin en daarop staat te maak vir besluitneming. Klein, medium en mikro-ondernemings (KMMOs) het nie die beskikbaarheid van DA- en ML-vaardighede in hul bestaande arbeidsmag nie, het beperkte infrastruktuur om ML te verwesenlik en het beperkte finansiering om ML-gereedskap en kundigheid aan te skaf. Hulle benodig behoorlike leiding oor hoe om ML op 'n lae-koste, haalbare en volhoubare manier aan te skaf.

Hierdie studie het gefokus om hierdie datagaping aan te spreek deur 'n besluitsteunraamwerk vir ML-algoritmes te skep. Die doel van hierdie studie was om 'n *besluitsteunraamwerk* te ontwikkel en te valideer, wat beide die *data eienskappe* en die *toepassingsdoel* oorweeg, om KMMOs in staat te stel

om die mees toepaslike ML-algoritme vir hul unieke data en toepassingsdoel te kies. Hierdie studie het gemik om die raamwerk vir 'n semi-vaardige ontleder, met met wiskunde, statistiek en programmeringsopleiding, wat vertroud is met die programmeringsproses, maar nog nie gespesialiseer het in die verskeidenheid van ML-algoritmes wat beskikbaar is nie, te ontwikkel.

Hierdie navorsingsprojek het die Sagtestelselmetodiek (SSM) gevolg en Jabareen se raamwerk-ontwikkelingsmetodologie gebruik. Verskeie literatuurstudies met betrekking tot data, DA, toepassingsdoeleindes, ML en die proses om ML toe te pas, was uitgevoer. Die KRuisIndustrie-StandaardProses vir Data-Ontginning (KRISP-DO) was gevolg om die eksperimente te ontwerp en te implementeer. Die resultate was geëvalueer en opgesom om die raamwerk vir besluitsteun te skep. Die raamwerk is bekragtig deur vakkundiges en moontlike eindgebruikers te raadpleeg.

# Contents

**Abstract** iii

**Opsomming** v

**List of Figures** xiv

**List of Tables** xvi

**Nomenclature** xx

**1 Introduction** 1

1.1 Background and motivation . . . . . 1

1.1.1 Types of data . . . . . 4

1.1.2 The data gap . . . . . 7

1.1.3 Small, medium and micro enterprises . . . . . 8

1.1.4 Existing decision support . . . . . 10

1.1.4.1 Existing data analytics software . . . . . 10

1.1.4.2 Existing frameworks for decision support regarding machine learning algorithms . . . . . 11

1.1.5 The focus of this research study . . . . . 16

1.2 Problem statement . . . . . 17

1.3 Research assignment . . . . . 17

1.4 Research objectives . . . . . 17

1.5 Research design . . . . . 18

1.6 Research methodology . . . . . 19

1.6.1 Soft Systems Methodology . . . . . 19

1.6.2 The research methodology for this research study . . . . . 22

1.7 Research scope, assumptions and limitations . . . 23
1.7.1 Scope . . . 23
1.7.2 Assumptions . . . 24
1.7.3 Limitations . . . 24
1.8 Ethical considerations . . . 25
1.9 The structure of the document . . . 26
1.10 Conclusion: Chapter 1 . . . 26

**2 Frameworks** **27**
2.1 Frameworks . . . 27
2.1.1 The definition of a framework . . . 27
2.1.2 Types of frameworks . . . 28
2.1.3 Types of conceptual frameworks . . . 29
2.1.4 Developing a framework . . . 30
2.2 The framework definition and framework development methodology for this research study . . . 32
2.2.1 The definition of a framework for this research study . . . 32
2.2.2 Comparing the research methodology and the framework development methodology . . . 33
2.2.3 The framework development methodology for this research study 34
2.3 Conclusion: Chapter 2 . . . 36

**3 Data and data analytics** **37**
3.1 Data analytics . . . 37
3.1.1 Types of data analytics . . . 37
3.1.2 The application purposes of data analytics . . . 39
3.1.3 Data analytics, data mining and machine learning . . . 42
3.1.4 The process of applying data analytics . . . 44
3.1.4.1 The CRoss-Industry Standard Process for Data Mining 44
3.1.4.2 The Sample, Explore, Modify, Model and Access process 52
3.1.4.3 The data analytics process for this research study . . . 54
3.2 Some characteristics of data . . . 55
3.2.1 The taxonomy of data . . . 55
3.3 Data preparation and preprocessing . . . 57

3.3.1 Data cleaning . . . 58
3.3.1.1 Outliers . . . 58
3.3.1.2 Errors . . . 59
3.3.1.3 Missing values . . . 61
3.3.2 Data transformation . . . 62
3.3.2.1 Numerical variables . . . 63
3.3.2.2 Categorical variables . . . 63
3.3.3 Normalisation . . . 64
3.3.4 Filtering . . . 64
3.3.5 Abstraction . . . 64
3.3.6 Reduction . . . 65
3.3.6.1 Data sampling . . . 65
3.3.6.2 Dimensionality reduction techniques . . . 66
3.3.6.3 Value discretisation . . . 66
3.3.7 Derivation . . . 67
3.3.8 Data division . . . 67
3.4 Conclusion: Chapter 3 . . . 69

**4 Machine learning** **70**
4.1 Machine learning . . . 70
4.2 Types of machine learning algorithms . . . 71
4.3 Classes of machine learning algorithms . . . 72
4.4 The selected machine learning algorithms . . . 77
4.4.1 Clustering algorithms . . . 78
4.4.1.1 Agglomerative hierarchical clustering . . . 78
4.4.1.2 Density-Based Spatial Clustering of Applications with Noise . . . 80
4.4.1.3 $k$-means clustering . . . 81
4.4.1.4 Mean shift clustering . . . 83
4.4.1.5 One-class support vector machine . . . 83
4.4.1.6 The advantages and disadvantages of the selected clustering algorithms . . . 83
4.4.1.7 The different clustering performance metrics . . . 85

4.4.2 Classification algorithms . . . 88
4.4.2.1 Decision trees . . . 88
4.4.2.2 $k$-nearest neighbour . . . 94
4.4.2.3 Logistic regression . . . 96
4.4.2.4 Naïve Bayes . . . 97
4.4.2.5 Neural networks . . . 98
4.4.2.6 Random forests . . . 101
4.4.2.7 Support vector machines . . . 103
4.4.2.8 The advantages and disadvantages of the selected classification algorithms . . . 108
4.4.2.9 The different classification performance metrics . . . 115
4.4.3 Regression algorithms . . . 117
4.4.3.1 Linear regression . . . 117
4.4.3.2 The advantages and disadvantages of the selected regression algorithms . . . 120
4.4.3.3 The different regression performance metrics . . . 120
4.5 Conclusion: Chapter 4 . . . 122

**5 Developing the decision support framework 124**
5.1 Developing the conceptual framework . . . 124
5.1.1 The basic idea for the decision support framework . . . 125
5.1.2 The five criteria of the framework . . . 126
5.2 Populating the conceptual framework I . . . 128
5.2.1 The datasets . . . 128
5.2.2 Data preparation and preprocessing . . . 129
5.2.2.1 Data cleaning . . . 129
5.2.2.2 Data transformation . . . 129
5.2.2.3 Normalisation . . . 130
5.2.2.4 Filtering . . . 130
5.2.2.5 Abstraction . . . 130
5.2.2.6 Reduction . . . 130
5.2.2.7 Derivation . . . 130
5.2.2.8 Data division . . . 131

5.2.3 Data type specific preparation and preprocessing . . . . . 131
5.2.3.1 Text data . . . . . 131
5.2.3.2 Image data . . . . . 131
5.2.3.3 Audio data . . . . . 131
5.2.3.4 Video data . . . . . 132
5.2.3.5 Transactional data . . . . . 132
5.2.3.6 Time-series data . . . . . 132
5.2.4 Building and implementing the models . . . . . 132
5.2.4.1 The machine learning algorithm implementations . . . 133
5.2.5 Problems encountered during the preliminary model deployment 136
5.2.5.1 Problematic machine learning algorithms . . . . . 136
5.2.5.2 Memory errors . . . . . 136
5.2.6 Evaluating the performance and execution time scores . . . . . 137
5.3 Populating the conceptual framework II . . . . . 137
5.3.1 Evaluating the programming scores . . . . . 137
5.3.2 Evaluating the interpretability and recommendation scores . . . 149
5.3.2.1 The text data . . . . . 149
5.3.2.2 Image data . . . . . 149
5.3.2.3 Audio data . . . . . 151
5.3.2.4 Video data . . . . . 151
5.3.2.5 Transactional data . . . . . 151
5.3.2.6 Time-series data . . . . . 153
5.4 The developed decision support framework . . . . . 153
5.4.1 An explanatory example . . . . . 154
5.4.2 The developed decision support framework . . . . . 155
5.5 Conclusion: Chapter 5 . . . . . 171

**6 Validation of the developed framework 172**
6.1 The subject matter experts for this study . . . . . 172
6.1.1 The application purposes . . . . . 174
6.1.2 The dataset preprocessing . . . . . 174
6.1.3 The machine learning algorithm implementations . . . . . 176
6.1.4 The five criteria of the framework . . . . . 178

6.1.5 General criticisms on the framework . . . . . 179
6.2 Possible end-users . . . . . 180
6.3 Conclusion: Chapter 6 . . . . . 183

**7 Research summary and conclusions** **184**
7.1 Project summary and conclusion . . . . . 184
7.2 Future research . . . . . 186
7.3 Appraisal of research work . . . . . 187
7.4 Concluding remarks . . . . . 188

**References** **215**

**A Summary of the datasets** **216**

# List of Figures

1.1 The project management triangle (Maynard, 2017) . . . . . 3
1.2 The data gap (Oosthuizen, 2018) . . . . . 4
1.3 The transformation of raw data to value (Tien, 2013) . . . . . 6
1.4 The relationship between data analytics and machine learning . . . . . 7
1.5 Value disciplines (Value disciplines image, 2018) . . . . . 15
1.6 McKinsey's strategic horizons (McKinsey's strategic horizons image, 2018) 15
1.7 Balanced scorecard (Balanced scorecard image, 2018) . . . . . 15
1.8 Ansoff matrix (Ansoff matrix image, 2018) . . . . . 15
1.9 The Soft Systems Methodology cycle of learning (Checkland & Poulter, 2006; Gasson, 1994) . . . . . 20

2.1 Jabareen's framework development methodology Jabareen (2009) . . . . 31

3.1 The types of data analytics (Mujawar & Joshi, 2015; Rajaraman, 2016) 39
3.2 The application purposes of data analytics . . . . . 40
3.3 Data analytics, data mining and machine learning . . . . . 44
3.4 The CRoss-Industry Standard Process for Data Mining (Nisbet *et al.*, 2009) . . . . . 45
3.5 The Sample, Explore, Modify, Model and Access process (Mariscal *et al.*, 2010) . . . . . 53
3.6 The taxonomy of data (Steynberg, 2016) . . . . . 56

4.1 The relationship between the four types of learning and the six classes of machine learning . . . . . 73
4.2 An example of a dendrogram of agglomerative hierarchical clustering . . 81

4.3 An example of the clustering results of Density-Based Spatial Clustering of Applications with Noise (Pedregosa *et al.*, 2011) . . . . . 82
4.4 The basic neural network (Neural netwotk image, 2018) . . . . . 99

5.1 The basic idea for the decision support framework . . . . . 125
5.2 The results when plotting image data . . . . . 150
5.3 The results when plotting audio features . . . . . 152
5.4 The results when plotting categorical data . . . . . 152
5.5 A guide for the developed decision support framework . . . . . 156
5.6 The framework section for the clustering of text data . . . . . 157
5.7 The framework section for the classification of text data . . . . . 158
5.8 The framework section for the clustering of image data . . . . . 159
5.9 The framework section for the classification of image data . . . . . 160
5.10 The framework section for the clustering of audio data . . . . . 161
5.11 The framework section for the classification of audio data . . . . . 162
5.12 The framework section for the clustering of video data . . . . . 163
5.13 The framework section for the classification of video data . . . . . 164
5.14 The framework section for the clustering of transactional data . . . . . 165
5.15 The framework section for the classification of transactional data . . . . 166
5.16 The framework section for the clustering of time-series data . . . . . 167
5.17 The framework section for the classification of time-series data . . . . . 168
5.18 The framework section for the regression of transactional data . . . . . 169
5.19 The framework section for the regression of times-series data . . . . . 170

6.1 The results of the three scenarios . . . . . 182

# List of Tables

1.1 Reconciling the Soft Systems Methodology and the research methodology for this study . . . 23

2.1 Comparing the Soft Systems Methodology and Jabareen's framework development methodology . . . 34

3.1 Textual categorical variable transformation (Nisbet *et al.*, 2009) . . . 63

4.1 A summary of types of clustering algorithms (Bijural, 2013; Moin & Ahmed, 2012) . . . 75
4.2 The three different output options of classification algorithms (Bijural, 2013) . . . 76
4.3 A summary of the applications of clustering algorithms . . . 79
4.4 Advantages and disadvantages of the selected clustering algorithms . . . 83
4.5 A summary of the applications of classification algorithms . . . 89
4.6 Advantages and disadvantages of the selected purely classification algorithms . . . 108
4.7 Advantages and disadvantages of the selected classification and regression algorithms . . . 109
4.8 A summary of the classification performance metrics for the different classification outputs (Pedregosa *et al.*, 2011) . . . 115
4.9 A summary of the applications of regression algorithms . . . 118
4.10 Advantages and disadvantages of the selected purely regression algorithms, namely linear regression . . . 120

5.1 The interpretation of the programming score . . . 127

5.2 The interpretation of the recommendation score . . . . . 128
5.3 The implementations of the machine learning algorithms in *Python* . . . 134
5.4 The programming score per application purpose and machine learning algorithm pair . . . . . 139
5.5 A small example for audio data . . . . . 154

A.1 Abbreviations used in the tables . . . . . 216
A.2 The text datasets used for clustering and classification . . . . . 218
A.3 The image datasets used for clustering and classification . . . . . 220
A.4 The audio datasets used for clustering and classification . . . . . 221
A.5 The video datasets used for clustering and classification . . . . . 223
A.6 The transactional datasets used for clustering and classification . . . . . 224
A.7 The time series datasets used for clustering and classification . . . . . 226
A.8 The transactional datasets used for regression . . . . . 228
A.9 The time series datasets used for regression . . . . . 230

# Nomenclature

### Abbreviations

| | |
|---|---|
| ADAM | Adaptive moment estimation function |
| AHC | Agglomerative hierarchical clustering |
| AI | Artificial intelligence |
| BFGS | Broyden–Fletcher–Goldfarb–Shanno solver |
| BNB | Bernoulli Naïve Bayes |
| CART | Classification and regression tree |
| CNB | Complement Naïve Bayes |
| CRISP-DM | CRoss-Industry Standard Process for Data Mining |
| DA | Data analytics |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise |
| DM | Data mining |
| DT | Decision tree |
| EF | Extra forest random forest by *Python* |
| ET | Extra tree decision tree by *Python* |
| FF-NN | Feed-forward neural network |

| | |
|---|---|
| FMI | Fowlkes-Mallows index |
| GNB | Gaussian Naïve Bayes |
| GPC | Gaussian process classifier |
| GTCA | Ground truth class assignments |
| ID3 | Iterative dichotomiser 3 |
| KD-tree | $k$-dimensional tree |
| KMC | $k$-means clustering |
| KNN | $k$-nearest neighbour classifier |
| KNR | $k$-neighbours regressor or $k$-nearest regressor |
| LBFGS | Limited memory Broyden–Fletcher–Goldfarb–Shanno solver |
| LDA | Linear discriminant analysis |
| LIBLIN | A library for large-scale linear classification imlemented by *Python* |
| LIN | Linear function |
| LinReg | Linear Regression |
| LogReg | Logistic Regression |
| MAE | Mean absolute error |
| MedAE | Median absolute error |
| MI | Mutual information-based score |
| miniKMC | mini-batch $k$-means clustering |
| ML | Machine learning |
| MLP | Multilayer perceptron |
| MNB | Multinomial Naïve Bayes |

| | |
|---|---|
| MSE | Mean squared error |
| MSLE | Mean squared logarithmic error |
| MS | Mean shift clustering |
| NB | Naïve Bayes classifier |
| NC | Nearest centroid classifier |
| NN | Neural network |
| NTCG | A truncated Newton method imlemented by *Python* |
| PEU | Possible end-user |
| POLY | Polynomial function |
| QDA | Quadratic discriminant analysis |
| R-NN | Recurrent neural network |
| RBF | Radial basis function |
| ReLu | Rectified linear unit function |
| RF | Random forest |
| RNN | Radius nearest neighbour classifier |
| RNR | Radius neighbour regressor |
| SAGA | A variant of SAG |
| SAG | Stochastic average gradient descent solver |
| SEMMA | Sample, Explore, Modify, Model and Access |
| SGD | Stochastic gradient descent solver |
| SIG | Sigmoid function |
| SME | Subject matter expert |

| | |
|---|---|
| SMME | Small, medium and micro enterprise |
| SSE | sum of squared errors function |
| SSM | Soft Systems Methodology |
| SVC | Support vector classification |
| SVM | Support vector machine |
| SVR | Support vector regression |

# Chapter 1

# Introduction

The aim of this research project is to apply engineering methods, skills and tools to develop and validate a decision support framework for machine learning (ML) applications in small, medium and micro enterprises (SMMEs).

In this chapter, the problem background, problem statement and the project motivation will be provided. Next, the project scope, assumptions, objectives and the problem solving methodology will be laid out. A preliminary literature study is also presented to serve as background and support for the research formulation. Finally, the structure of the report and the conclusions of this chapter will be detailed.

## 1.1 Background and motivation

Data is currently one of the most critical and influential emerging technologies. *Big data* is the term which is used to explain the current explosion of the great variety of data types which are created from different sources. Big data refers to large *volumes* of data with increased *variety*, *velocity*, *veracity* and *value* (Corrigan *et al.*, 2012). The digital universe, the measurement of all the digital data created, replicated and consumed in one year, doubles every two years from now until 2020, according to Gantz & Reinsel (2012). Organisations and employers around the globe strive to investigate and exploit the exponential data growth to discover hidden insights in an effort to create value (Zhu *et al.*, 2014).

The goals of utilising data in organisations in an effort to increase value creation are as follows (Zhu *et al.*, 2014):

1. Revenue

   The analysis of data is used to create new revenue streams, explore new business models, increase revenue, reduce cost, enable process optimisation, increase operational efficiencies and productivity, increase quality, reduce risk and manage data at low cost.

2. Customer service

   The analysis of data is used to enable a greater understanding of customer needs. Through understanding their customers' behaviours, preferences and needs, organisations can improve and customise their products and services, retain and gain customer loyalty, follow market trends, improve their marketing techniques and improve their competitive performance in the market.

3. Business development

   The analysis of data is used to enable decision support and faster decision making, improve employee morale and productivity, create new product or service offerings, enable the outsourcing of non-core activities and functions, support decisions regarding mergers and acquisitions, enable divestitures, gain competitive insight, increase organisation agility and government, ensure regulatory compliance and reduce risk.

These goals can be summarised using the project management triangle or iron triangle, as illustrated in Figure 1.1. It indicates the three main goals of any project, namely reduce cost, reduce time and increase quality. These goals are also constraints in the project and generally there are trade-offs where the focus can only fall on two of these goals at a time (Maynard, 2017).

Data is readily available everywhere and this explosive data growth is driven by a variety of factors, including (Gantz & Reinsel, 2012):

- The decreasing technology costs of devices which create, capture, manage, process and store data.


Figure 1.1: The project management triangle (Maynard, 2017)

- The cost of data storage, processing and bandwidth has decreased significantly while the data quality, network access, computational capacity and the availability of more powerful data analytic tools have increased significantly (Corrigan *et al.*, 2012).
- The increasing availability and usage of the internet, communication platforms, multi-media and social media platforms. In addition, human behaviour is captured in the forms of visual, audio and textual data on these platforms.
- The increasing availability of machine-created data.
- The growth of meta-data and meta-information, *i.e.* information about information.
- Increasing use of automation, robotisation and the Internet of Things (IoT) (Zhu *et al.*, 2014).

With this explosive data growth, new problems arise, for example, the required capabilities for the processing of the data. Other problems include determining rules to dictate the use and distribution of the data or gathering the appropriate skills and expertise to manage and analyse the data as well as interpret the results of the analytics (Zhu *et al.*, 2014).

There is a data gap. Data is readily available and easy to capture; however, the information therein remains untapped yet ready for digital explorers to discover the hidden value in the data. The true potential of data is yet to be exploited since,

currently, about 1% of generated data is ever actually analysed (Gantz & Reinsel, 2012). Figure 1.2 illustrates this data gap. Little value creation takes place due to various reasons, for example, the data creators are unaware of the potential their data holds. The prominent reason is that there is a data skill gap which is a hindrance to the process of creating value from data (Zhu *et al.*, 2014).

### 1.1.1 Types of data

Data may be categorised according to different characteristics. One possible category is the different data structures and formats during the life cycle of data. Based on this characteristic, the following three data types are identified:

1. Structured data

   Structured data is data which is converted to a common format and organised in tables with keys to link them together to indicate relationships in the data (Rajaraman, 2016). Examples are organised, integrated and relational databases used in businesses to perform their services and it is typically managed by software such as Oracle or a query language like SQL (Nisbet *et al.*, 2009).

2. Semi-structured data

   Semi-structured data is a form of structured data which is in-between a formal, relational database and loose, unrefined and disorganised data. It contains relational indicators to enforce separation and hierarchies within the data.

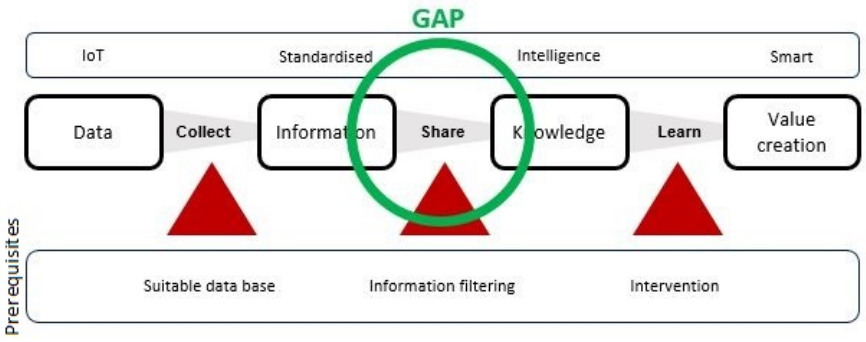


Figure 1.2: The data gap (Oosthuizen, 2018)

3. Unstructured data

   Unstructured data is data which is created by various sources and is not yet aggregated and integrated. Relationships therein are not available or indicated (Nisbet *et al.*, 2009) and it is not organised in a pre-defined manner. Examples are notes, memos and reports used in a business as well as social media data, including e-mails, tweets, blogs, websites and Facebook posts (Rajaraman, 2016).

Gantz & Reinsel (2012) defines data technologies as a new, advanced generation of technologies, techniques, structures and architectures which is designed to efficiently and effectively extract value from large volume, wide variety datasets by allowing high-velocity access, capture, analysis and discovery.

In order to gain value from data it has to be processed and analysed. The results have to be interpreted to be able to extract information from it to enable decision making support across a wide range of areas, including business, technology, science, engineering, education, healthcare, environment and the society at large (Tien, 2013).

Very large datasets are known as big data. In 2013, a dataset was classified as big data if its size ranged between terabytes ($10^{12}$ bytes) and pentabytes ($10^{15}$ bytes). However, as software tools and technology (greater storage capacity and improved central processing units (CPUs) in computers) become more powerful, this definition will be adjusted accordingly (Tien, 2013). It was determined that in the year 2017, 26 zettabytes ($10^{21}$ bytes) of data was generated and it was predicted that in the year 2019, 41 zettabytes of data would be generated world wide (Holst, 2017); thus, the range of size increased and the definition of big data was slightly adjusted (Rajaraman, 2016).

The characteristics of data are (Rajaraman, 2016; Zhu *et al.*, 2014):

1. Volume

   Volume refers to the size of the dataset, measured in bytes. Typical sizes are in the order of zettabytes.

2. Variety

   Variety refers to the diversity in the data. As technology develops, a greater variety of data sources are created and the types of data available increase. For example, text and numerical data expanded to include image, audio and video

data. Additionally, various new data structures are created and the life cycle of data expands to accommodate the additional changes in data structure and format.

3. Velocity
   Velocity refers to the rate of change in data. Traditionally, data changed slowly. In the modern world the data is created in real-time and changes quickly.

4. Veracity
   Veracity refers to the quality and trustworthy characteristics of the data. Modern data contains more noise, biases, insignificant values, errors and inconsistencies which impact the quality thereof and influence its statistical measurements, for example, standard deviation.

5. Value
   The data is raw and needs to be processed to be converted into information. Knowledge is extracted from the information and value is derived from the knowledge. This value extraction process is illustrated in Figure 1.3. A synonym for value is the 'visibility' of the data.

The growing variety in data structures, life cycle and characteristics add additional components which need to be considered when performing the process of extracting information and value from data.

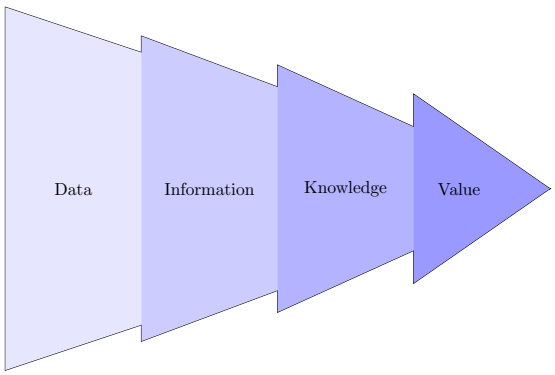


Figure 1.3: The transformation of raw data to value (Tien, 2013)

### 1.1.2 The data gap

Value creation from data is made possible through data analytics (DA). In a survey of 437 companies, 76% indicated an interest to invest in ML technologies over the years of 2015 to 2017 (Kart & Heudecker, 2015). However, in a survey of 400 companies, 4% indicated that they have the right strategic intent, skilled people, resources and data to gain meaningful insights from their data and to act on them (Sinha & Wegener, 2013). According to Kart & Heudecker (2015), the successful adoption of ML depends on finding talented data scientists who can execute the technology as well as understand its strengths, weaknesses, pitfalls and limitations. According to Economist Intelligence Unit (2014), 43% of North America C-suite level managers think their senior management colleagues lack necessary skills or expertise in utilising data for decision making purposes.

One main factor contributing to this gap is the lack of expert knowledge in the field of DA. *Data analytics* is the process of investigating and exploring data to derive insightful and relevant trends and patterns which can be used for a wide variety of applications, including decision support and process optimisation (Zhu *et al.*, 2014). The need for DA is growing, since 33% of business leaders distrust the information they use to make business decisions (Zhu *et al.*, 2014).

Machine learning is most widely used to perform DA and it is a subset of DA, as illustrated in Figure 1.4. *Machine learning* consists of algorithms (sets of rules) that employ mathematical and statistical techniques to give computers the ability to study, learn from, identify trends and patterns, and determine similarities in data. Machine learning algorithms infer their own rules from experience (the data) instead of following a set of step-by-step rules as in traditional programmed algorithms (Thurn & Anderson, 2017).

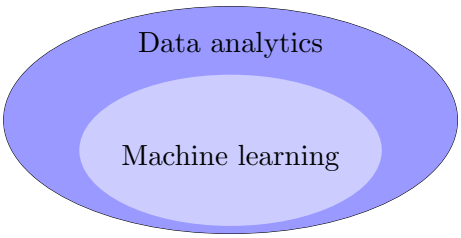


Figure 1.4: The relationship between data analytics and machine learning

A great variety of ML algorithms which are designed for processing large-scale and high dimensional data with noise, with promising efficiency and accuracy are available. Choosing an algorithm for an application is difficult without prior knowledge of all the available algorithms. Choosing an insufficient algorithm can compromise the results and may result in poor decision making. Generally, in an ML experiment, various algorithms are considered with different iterations of each since parameter tuning is critical to the performance of the ML algorithm. After the algorithms or models have been trained, statistical comparison tests are conducted to identify which model performed best. This model is then chosen as the final model for all future work. This process is time-consuming since it requires the data scientist to study the various algorithms, implement various versions, and conduct tests to choose the appropriate algorithm. An alternative would be to hire a data analyst or scientist; however, this may result in a costly venture. Another alternative could be the acquisition of DA and ML software which has a variety of benefits and drawbacks. Another resort is to use an available decision support framework to help identify the appropriate algorithm to select and use. To understand the target user of the framework, it is briefly discussed next.

### 1.1.3 Small, medium and micro enterprises

Small to medium enterprises or the latest term, small, medium and micro enterprises (SMMEs) can be qualitatively described as specific enterprises which are characterised as the drivers of national economic growth and creators of opportunities with the largest potential of (self-) employment. They are also characterised as generators of new jobs and influencers of national, regional and local development (Spicer, 2006). They have international character since they also perform business globally. They are key drivers of economic growth, innovation and job creation (SEDA, 2016).

SMMEs are classified according a few quantitative metrics: the annual turnover, the number of paid employees and total gross asset value. The annual turnover can further be divided per economic sector, including mining, manufacturing, construction, transport and retail trade. There are specific values of each of the three metrics which are used to separate and classify SMMEs.

The characteristics of SMMEs include the following (Fink & Kraus, 2009; SEDA, 2016):

1. Identify opportunities and business ventures
   Starting a business is a risky undertaking since a new course of action is attempted. SMMEs identify solutions to problems or discover new ideas, which are transformed into businesses opportunities and ventures.

2. Innovative and flexible
   SMMEs create new methods, products or service delivery through innovative approaches. They also have to use innovation to readily and quickly adapt to changing or new circumstances or environments.

3. Enable job creation
   SMMEs are new entrants to markets and create jobs to realise their product and service offerings.

4. New entrants to the market
   SMMEs provide innovative or new methods, products and service delivery to existing markets.

5. Exposed to higher risk and small growth rates
   Businesses are threatened by internal and external forces. Since SMMEs have limited resources, infrastructure, knowledge and experiences compared to established businesses, they are exposed to more risks and higher risks than established businesses. Due to these factors, SMMEs also experience small growth rates.

6. Low survival rates
   According to Burns (2016), only 50% survive after their 5th year in the European Union. Since SMMEs have limited resources their chances of surviving the higher risks which they are exposed to are low.

SMMEs want to create value by performing margin management to increase effectiveness and efficiency. They want to enable asset growth to increase market penetration and to fulfil investor expectations in an attempt to gain access to finance and to attract more investors.

However, SMMEs are exposed to challenges, including (SEDA, 2016):

- Access to finance and credit.
- Poor infrastructure.
- Low levels of research and development capacity.
- Inadequately educated workforce.
- Lack of access to markets.

As previously stated, value creation from data is made possible through DA by utilising ML. In summation, SMMEs whose core business is not DA, lack the availability of DA skills in their existing workforce, have limited infrastructure to realise ML and have limited funding to employ DA tools and expertise. These drawbacks limit their capability to perform DA and ML. They need proper guidance as to how to employ ML in a low-cost, feasible and sustainable way.

### 1.1.4 Existing decision support

A preliminary literature study was performed by the researcher to determine the appropriateness of existing decision support for DA with ML for SMMEs whose core business is not DA. Existing DA software, academic decision support frameworks and frameworks implemented in practice were investigated. The hiring of data scientists was ignored since it is considered too costly and is preferred that the developed applications remain and are managed in-house.

#### 1.1.4.1 Existing data analytics software

Various DA software has been developed by different companies around the globe, for example, Sisense, IBM Watson, Looker, Yellowfin and many more.

IBM has developed an IBM Big Data and Analytics platform for three types of users: business users, developers and administrators. The platform enables business users to explore and visualise data, and developers are given access to various DA methods; however, it does not give an indication of what DA and ML algorithms would be best to perform on their unique data (Zhu *et al.*, 2014).

The software Sisense is applicable to both data scientists and business users and has various features, including personalised dashboards, interactive visualisations and analytical capabilities (including the use of ML algorithms) (FinancesOnline, 2019).

The benefits of the existing DA software include:

- Interactive data visualisation and personalised dashboards.
- Natural language detection technology and anomaly detection methods.
- Non-technical users with no programming background can use the software since it simplifies DA and requires no hard coding and aggregating modelling.
- Descriptive, predictive and prescriptive analytics are included.
- Web integration and high system security.
- Accessible data and data scheduling.
- Insightful reports and collaboration (FinancesOnline, 2019).

The drawbacks of these software from the perspective of SMMEs include:

- It is an expensive investment, especially for SMMEs.
- It is an elaborate system, where only a small percentage of its functions are applicable. Thus, the software is over-designed for SMMEs.
- SMMEs might be overwhelmed by the complexity of the software and its implementation for use.

#### 1.1.4.2 Existing frameworks for decision support regarding machine learning algorithms

Few frameworks which aid in choosing an appropriate ML algorithm given the available data exist. Also, it seems the definition of framework varies both in academic literature and practical work.

1. Framework definition in academic literature

   A variety of frameworks were found in literature, indicating that the definition thereof varies. The definition also varies per industry or application domain. For

example, work produced in the ML domain provides algorithms to address very specific problems in the domain, including learning from dense data sets (Mirhoseini *et al.*, 2018) and graph based semi-supervised learning (Pei *et al.*, 2017) whilst work published in the manufacturing and production management industries have a greater spectrum, including diagrams, algorithms and manuals. Some work introduce diagrams with logical flows and visual mapping of processes with decision making nodes or options (Barreiro *et al.*, 2003; Spinler & Kretschmer, 2013). Others provide step-by-step rules or algorithms to implement decisions in a logic or mathematical sequence (Balcik & Ak, 2014; Chen *et al.*, 2010). Some give a broad outline of factors to consider in a process (Abrahams *et al.*, 2015). Some work provide a comprehensive document requiring thorough reading similar to a manual (Criminisi *et al.*, 2012). More general frameworks are available, for example, general information for decision forests (Criminisi *et al.*, 2012) or possible unknown unknowns in project management (Ramasesh & Browning, 2014), while others provide support for problem specific situations, for example, a school feeding supply chain framework (Spinler & Kretschmer, 2013) and an error adjustment in machining (Wan *et al.*, 2008).

A thesis by Balcan (2008) introduced a new general model for semi-supervised learning and developed algorithms with better guarantees than those developed at the time. It specialises only in one type of learning (semi-supervised learning) and does not aid the user in choosing the appropriate algorithm given their data and application purpose. It is limited to few application options: clustering algorithms and active learning. Also, it does not provide a clear diagram with logical flows to aid the user to reach a decision on an algorithm; instead a document is written according to the applications which are possible.

In Gorban *et al.* (2018), a conceptual framework is proposed for augmenting artificial intelligence (AI) in communities or social networks of AI. Again no resulting diagram is presented, although theorems have been proven. This work does not aid in assisting with the decision to choose an appropriate algorithm given data and an application purpose.

Criminisi *et al.* (2012) provides a framework for decision forests for a variety of applications, including classification, regression, density estimation, manifold learn-

ing, semi-supervised learning and active learning. New and efficient algorithms are proposed as well. No overall or summation diagram is presented. Instead the document is written according to the applications which were available.

Paredes (2018)'s thesis introduced a framework to guide organisations on integrating ML into their enterprise, focusing on the enterprise model, opportunities, technological adoption and ML systems' architecture. A clear diagram with logical flow is presented with a thorough discussion on how to interpret it.

Little academic literature was found on ML frameworks on the Stanford University, Cambridge University and Oxford University repositories. In the IEEE Transactions on Neural Networks and Learning Systems repository a few articles on frameworks were found; however, they were focused on developing computing frameworks or algorithms which perform specific tasks to address specific problems in the ML domain (Chen *et al.*, 2018; Niu *et al.*, 2018).

2. Framework definition in practice

   A wide variety of frameworks in practice were found. There is a difference between frameworks used in data science and frameworks used in business. Few frameworks on decision support regarding ML algorithms were found, especially in terms of business, although the researcher thought it important to review the types of frameworks implemented in practice to gain an idea of what frameworks are from the perspective of businesses. There are also different types of frameworks within business, including strategy or strategic frameworks, frameworks for internal analysis, frameworks for external analysis, business frameworks and information technology frameworks. Some businesses use frameworks which have been developed in academia or based on research for general applications and others developed personal frameworks specialising in their industries or fields.

   KDnuggets is a leading online platform on AI, DA, data, data mining (DM), data science and ML. KDnuggets uses the word "framework" in a similar fashion as libraries or packages which are available to use for the application of ML algorithms in programming languages (Desale, 2016).

   Babuta *et al.* (2018) released a report which provides information on the policing and risks of ML algorithms and include characteristics like discretion, accountability, transparency, intelligibility, fairness and bias within this ML policing.

Sapp & Gartner Inc. (2017) published a document explaining what ML is, how it benefits an organisation, how a business should prepare for ML and how to get started with ML. Both these documents are like manuals instead of diagrams with logical flows and should be read thoroughly before implementation. Neither assists the user in selecting an appropriate ML algorithm given their requirements.

According to Muehlhausen (2012), businesses sometimes confuse business models, frameworks and architectures and use them interchangeably. A business model presents the rationale of how a business creates, delivers and captures value. A business framework describes the management structure, corporate organisation, company policies or the method used to achieve a particular goal, including the policy, procedure and management changes incorporated by it. A business architecture is based on corporate business and presents documents and diagrams which describe the structure of the business in terms of functionality, services and information.

Strategic frameworks assist in identifying goals and help the business to stay focused thereon. According to Wright (2018), the top five strategy frameworks are McKinsey's strategic horizons, value disciplines, the stakeholder theory, the balanced scorecard and the Ansoff matrix. Other strategy frameworks include Maslow's hierarchy as a business framework and the VRIO framework. Some frameworks were developed by observing and researching trends in companies, including McKinsey's strategic horizons (developed from research by consultants from McKinsey & Company (Hill, 2017)) and value disciplines (created by Michael Treacy and Fred Wiersema after researching trends in companies). Others are more theoretically based, including stakeholder theory, which is based on the assumption that when value is delivered to the majority of a business's stakeholders, then the business is considered successful. The balanced scorecard is also theoretical in nature. It was developed to measure performance using a balanced set of performance measures and it has evolved to a fully integrated strategic management system (Balanced Scorecard Institute, 2019). The balanced scorecard, value disciplines, stakeholder theory and Ansoff matrix strategic frameworks assist businesses in identifying areas to focus on for improvement. McKinsey's framework presents a process over time which can be followed to improve business

offering. Figures 1.5 to 1.8 illustrate the following strategic frameworks in order: the value disciplines, McKinsey's strategic horizons, the balanced scorecard and the Ansoff matrix (Wright, 2018).

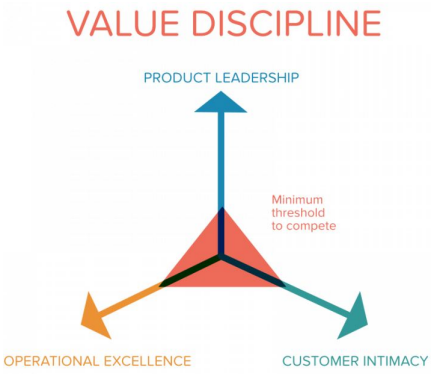


Figure 1.5: Value disciplines (Value disciplines image, 2018)

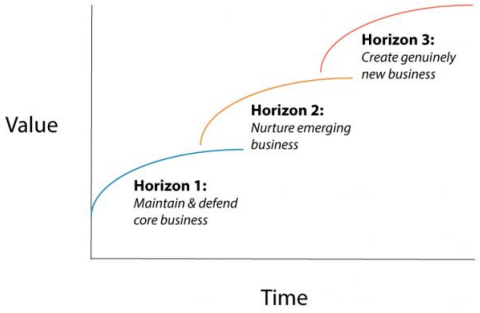


Figure 1.6: McKinsey's strategic horizons (McKinsey's strategic horizons image, 2018)

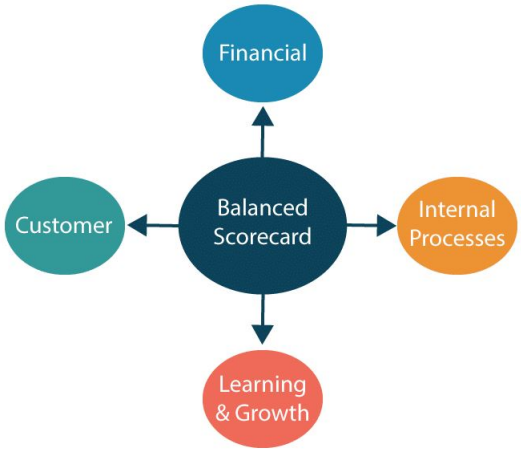


Figure 1.7: Balanced scorecard (Balanced scorecard image, 2018)

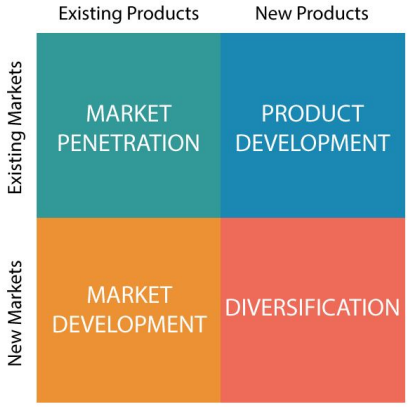


Figure 1.8: Ansoff matrix (Ansoff matrix image, 2018)

According to Taylor & Mariton (2012), there are two types of frameworks or tools depending on the application area of the business, namely frameworks for internal analysis and frameworks for external analysis. Internal analysis frameworks are applied to the business itself, to assess and change factors the business can control.

It includes the strengths, weaknesses, opportunities and threats (SWOT) analysis; value chain analysis; the business model canvas; the balanced scorecard; VMOST; resources based view (VRIN Model) and Kotter's change model. Most of these tools help identifying areas which can be addressed to improve business. VMOST has added structure since it provides a hierarchy of steps to help the business align to its strategy. Kotter's change model provides a logical flow of sequential steps to incorporate change in the business (Taylor & Mariton, 2012).

External analysis frameworks are applied to the business environment outside the business itself, to assess and react to factors the business has no control over. It includes scenario planning and Porter's five forces. Scenario planning has added structure since it provides four sequential steps of creating and managing scenarios (Taylor & Mariton, 2012). Porter's five forces include the evaluation of the following five forces or threats: competition from the industry, the threat of new players in the industry, the power of suppliers, the power of the customers and the threat of substitutes. Another tool for external analysis is PESTLE, which describes the political, economic, social, technological, legal and environmental factors. These tools help to identify, assess and manage external factors.

### 1.1.5 The focus of this research study

Hiring data analysts or acquiring DA software is costly, especially for an SMME which is entering the world of DA and wants to start with small projects. Few frameworks are available to assist in choosing an appropriate ML algorithm given the data characteristics and application purpose of a problem. With their limited knowledge to apply an appropriate ML algorithm, inappropriate algorithms might be selected and developed, leading to insufficient results and poor decision making support. Consequently, the quality of their product and service offering might be affected and organisational costs might increase. Although it is relatively inexpensive to capture and collect data, assistance is needed to enable value creation from the data.

## 1.2 Problem statement

Given the previous arguments and findings of the preliminary literature study, it was found that a gap exists in the data analytics and machine learning capabilities of small, medium and micro enterprises. Therefore, the aim of this research study is to develop a *framework* for selecting machine learning algorithms to support small, medium and micro enterprises to choose the appropriate algorithm given the *data characteristics* and *application purpose*. This study aims to develop the framework for a semi-skilled analyst with mathematics, statistics and programming education, at least on undergraduate level, hereafter termed 'analyst'. The analyst is familiar with the process of programming, yet has not specialised in the variety of machine learning algorithms which are available. The analyst typically works at a developing small, medium and micro enterprise with limited resources, including time, money and computational power. The idea is to assist the analyst in choosing the appropriate algorithm whilst considering limiting factors, for example, minimal time and cost implications. The study will use programming languages which are freely available and well supported to enable cost-savings. The trade-offs of the project management triangle will also be indicated in the framework, in terms of computational cost, execution time and performance quality to further support decision making whilst considering the limited resources available.

## 1.3 Research assignment

Given the problem background and motivation, the research assignment can be stated as:

**Develop** and **validate** a *decision support framework* which considers both the *data characteristics* and the *application purpose* to enable small, medium and micro enterprises (SMMEs) to choose the appropriate machine learning (ML) algorithm for their unique data and application purpose.

## 1.4 Research objectives

The research problem as stated will be solved by pursuing the following sequential objectives:

1. *Gain* knowledge and understanding of data, DA and ML algorithms.

2. *Develop* a decision support framework which provides a variety of ML algorithms given the characteristics of the data and the purpose of the application.

3. *Expand* the framework to such an extent that it indicates the appropriate ML algorithm per data characteristic and application purpose pair, whilst considering the project management triangle. For example, provide the ML algorithms in descending order of performance quality and in ascending order of the required execution time.

4. *Expand* the framework to such an extent that it indicates the relative trade-offs of the iron triangle per ML algorithm application.

## 1.5 Research design

Before the research methodology can be identified, the research design must be determined. The research design aids in directing the study and identifying methodologies and methods needed to realise the project objectives. Three different research designs are available in literature: quantitative, qualitative and mixed-method designs (Bryman *et al.*, 2017). Quantitative methods focus on collecting numbers while qualitative methods focus on collecting texts or words. Mixed-methods designs are a combination of quantitative and qualitative methods (Greene *et al.*, 1989). They are briefly described below.

1. Quantitative design
   The quantitative design determines, describes and analyses the relationships and correlations between variables by collecting and examining numeric data represented by numbers, scores or statistical values (Bryman *et al.*, 2017; Plano Clark & Ivankova, 2016). The data is mainly collected by using instrument based experiments or observations and by gathering performance data from real world events (Creswell, 2003). Surveys can also be utilised.

2. Qualitative design
   The qualitative design determines, describes and analyses individuals' experiences by collecting and examining narrative data represented by spoken words, text or images (Bryman *et al.*, 2017; Plano Clark & Ivankova, 2016). The data is

mainly collected by interviews and surveys with open-ended questions so that participants' views can be expressed. The process is more inductive in nature (Creswell, 2003). Case studies and narrative research can also be utilised.

3. Mixed-methods design
   The mixed-methods design integrates quantitative and qualitative methods of data collection and examination in a process of understanding and conceptualising a research purpose (Plano Clark & Ivankova, 2016). It employs a variety of data collection methods, including surveys, experiments and observations of real world events. It makes use of various data analysis techniques, including statistical or textual and image analysis (Creswell, 2003). By combining both designs, the findings and conclusions of the study are more complete and justifiable compared to only using one design to address the study (Bryman *et al.*, 2017).

For this research project, the mixed-methods design will be implemented, since both experiments (quantitative methods) and interviews with subject matter experts (qualitative methods) will be utilised to develop and validate the decision support framework.

## 1.6 Research methodology

In order to meet the project objectives, a certain methodology must be followed. The methodology used is dictated in part by the research rationale. The research methodology of this study follows the Soft Systems Methodology (SSM).

### 1.6.1 Soft Systems Methodology

Checkland (1981) developed the SSM as a method or technique for investigating an unstructured problem with a weak defined problem situation which requires thorough contextual understanding. It aids in structuring the problem and discovering a solution to the problem by enabling problem identification, model building, situation analysis and action implementation (Checkland, 2000; Checkland & Poulter, 2006). The SSM can be used for theory generation as well as theory testing. The SSM is summarised in Figure 1.9.

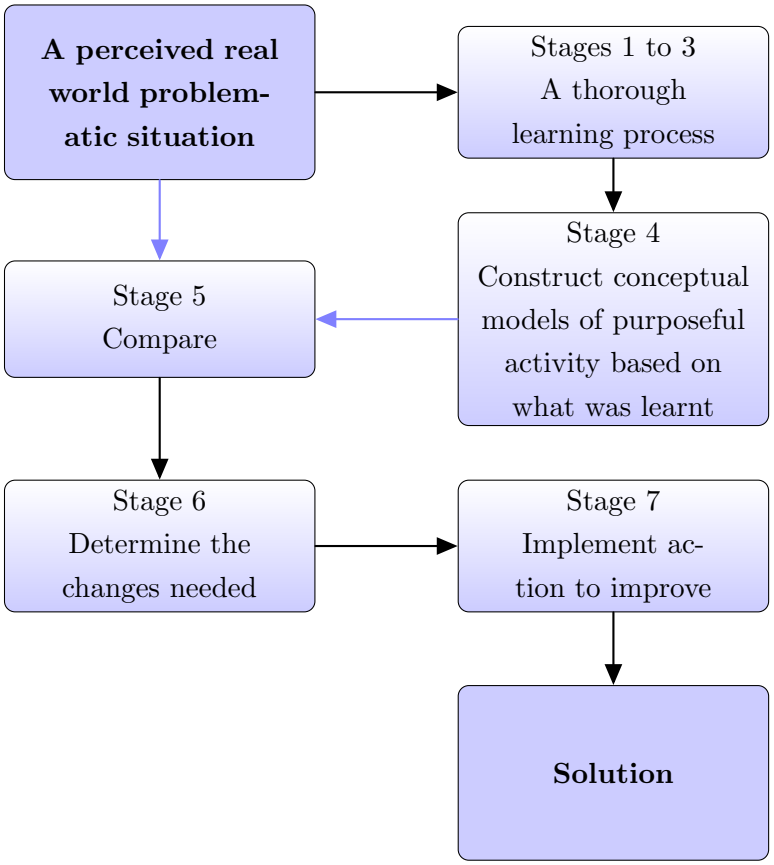


Figure 1.9: The Soft Systems Methodology cycle of learning (Checkland & Poulter, 2006; Gasson, 1994)

The SSM consists of the following seven stages (Checkland, 1981; Gasson, 1994):

1. Consider the variation of the problem situation
   The problem situation is investigated to determine the context and content thereof. The aim is to determine a holistic view of the problem. Stage 1 is a prelude to the following stage since it facilitates the progression to a state where the situation is understood and the capability to express it in words and diagrams is available (Gasson, 1994).

2. Express the problem situation
   Express the problem situation explicitly in text and images to further the understanding of the problem situation. A thorough investigation is needed to ensure

that as much information as possible is included and conveyed to present a complete, wider-ranging expression of the problem situation. The need to address the problem situation and the scope of the solution have to be illustrated. The objectives of the solution should be presented as well (Checkland & Poulter, 2006; Gasson, 1994).

3. Define significant and purposeful root definitions
   Determine the environment that the problem situation operates in and identify the stakeholders of the problem situation. While considering interests of stakeholders, determine questions (and their answers) to help identify concepts which describe what is happening in the problem situation. The goal is to name the problem situation which facilitates the understanding of the problem situation (Checkland & Poulter, 2006).

4. Construct conceptual models
   The solution to the problem situation is presented in the form of a conceptual model. The conceptual model is build by identifying and analysing all the activities needed in order to clearly define what must be done to achieve the solution. The activities should only be applicable to the solution itself and should achieve the desired objectives of the solution. The activities are listed in sequential order. Activities which monitor the solution development process and present feed-back results should be included. The combination of these activities provides the solution, the conceptual model, to the problem situation (Checkland, 2000).

5. Compare the conceptual model and the problem situation
   The developed conceptual model is compared to the real world to determine a list of changes which needs to be implemented in order to move the problem situation to the one modelled in the conceptual model.

6. Construct feasible and desirable changes
   The identified changes are evaluated in terms of feasibility and desirability from the perspective of the stakeholders and the available resources. The impact on the stakeholders, should the changes be implemented, is considered as well (Checkland & Poulter, 2006).

7. Define and implement action to improve the problem situation
   The changes should be implemented with two goals in mind, namely minimising the impact on the stakeholders and achieving the objectives of the solution.

### 1.6.2 The research methodology for this research study

Given the SSM research methodology and research objectives in the previous sections, the SSM was adapted to formulate the research methodology for this research study. It was adapted to suit the needs of this research study and to ensure that the research requirements and objectives are met throughout the process. For this research study, the problem situation is the need for proper guidance regarding the implementation of ML algorithms and the conceptual model or solution is a decision support framework for ML applications. The research methodology for this research study comprises of the following phases:

1. *Perform* a literature study on frameworks and methodologies for framework development to determine which type of framework to develop for this research study as well as how to develop the framework.

2. *Perform* a literature study on data and DA to gain an understanding of the types of data and DA available and used in ML applications.

3. *Perform* a literature study on the available ML algorithms, including their categorisation, methodologies, advantages, disadvantages, application purposes and performance evaluation measurements to gain an understanding of ML algorithms and their application purposes.

4. *Develop* an appropriate basic conceptual decision support framework.

5. *Design*, *perform* and *analyse* experiments with relevant data to quantitatively evaluate the performance of the ML algorithms in terms of the data characteristics, application purposes and the iron triangle.

6. *Consult* applications in literature and practice to further detail the framework.

7. *Expand*, *improve* and *validate* the technical and quantitative aspects of the decision support framework by consulting subject matter experts (SMEs).

8. *Validate* and *improve* the user-friendliness of the decision support framework by consulting possible end-users (PEUs) in the engineering field.

9. *Provide* the project conclusions based on the results.

The correlations between the SSM and the research methodology for this research study is presented in Table 1.1. In terms of the SSM, the real world provides a problem situation and the solution is implemented to change the real world. In this research study it is reversed. In terms of this research study, the real world provides the solution (what the study aims to model) and the framework is the problem situation (the framework should be developed and adjusted to reflect the real world).

Table 1.1: Reconciling the Soft Systems Methodology and the research methodology for this study

| **Soft Systems Methodology** | **Research methodology** |
|---|---|
| 1. Consider the variation of the problem situation<br>2. Express the problem situation<br>3. Define significant and purposeful root definitions | 1. Literature study on frameworks<br>2. Literature study on data and DA<br>3. Literature study on ML algorithms |
| 4. Construct conceptual models | 4. Develop a basic conceptual framework<br>5. Perform experiments<br>6. Consult applications in literature |
| 5. Compare the conceptual model and the problem situation | Validate the framework:<br>7. Consult SMEs<br>8. Consult PEUs |
| 6. Construct feasible and desirable changes | 9. Provide conclusions |
| 7. Define and implement action to improve the problem situation | |

## 1.7 Research scope, assumptions and limitations

The following section will detail the scope, assumptions and limitations of the study.

### 1.7.1 Scope

Given the research goal of this research study, the scope of this study is as follows:

- The research will be limited to unsupervised, semi-supervised and supervised learning algorithms. It will exclude reinforcement learning as it is a time-consuming process requiring specific programming and multiple simulations.

- Dimensionality reduction techniques will not be covered.
- The ML algorithms used in this research study will be limited due to scope reasons. The selected ML algorithms will be identified later in the study.
- The ML algorithms will be implemented in *Python* using the build-in *sklearn* libraries and packages.
- The ML algorithms will be implemented in their recommenced states as provided in *Python* and the researcher will not experiment with each algorithm until the best version of the algorithm has been discovered, *i.e.* parameter tuning will not be included in this work.

### 1.7.2 Assumptions

Given the research goal of this research study, the assumptions applied in this study are as follows:

- The framework will be developed for a semi-skilled analyst with limited knowledge of ML algorithms, including their benefits, drawbacks, pitfalls and limitations.
- The framework will be designed for users with limited resources, including the computational power available and finances to acquire additional computational power or DA software.
- Data dimensionality reduction techniques will not be needed since it is assumed that the dimensionality of the data will be controllable.
- The datasets utilised in the experiments are representative of the real world.

### 1.7.3 Limitations

The limitations in this research study are as follows:

- Limited computation power will be available. Four identical computers with 12 GB RAM memory, i7 core and *Windows 10* will be utilised for the ML implementations and experiments.
- Due to time constraints a total of 90 datasets were identified, preprocessed and utilised.

## 1.8 Ethical considerations

Ethical clearance to consult SMEs and PEUs was obtained from Stellenbosch University. The following ethical considerations were applicable to the SMEs and PEUs consulted during the research study:

- Written consent was requested from the SMEs and PEUs before the interviews or consultations. The consent form provided the necessary information as to what was required from them.
- The SMEs were consulted for their technical expertise to provide technical evaluations and validations of the research study.
- The PEUs were consulted to provide evaluations of the developed decision support framework in terms of the usefulness, user-friendliness, applicability and interpretability thereof.
- To provide validity, reliability and credibility to the research study, the titles, initials, surnames and professions of the SMEs and PEUs are available in this document.
- The no other personal information was collected, requested or used.
- The SMEs and PEUs had an option to remain anonymous should they decide to do so and all information would have been treated confidentially. If they decided to remain anonymous, they would not have been treated differently or be impacted negatively.
- The SMEs and PEUs also had the option to withdraw from the research at any time and they would not have been treated differently or be impacted negatively. Theses SMEs and PEUs were not contacted again.

The ethical clearance document and the written consent forms are available from the researcher and may be provided on request.

## 1.9 The structure of the document

The research document is structured according to the research methodology presented in Subsection 1.6.2.

**Chapter** 2 reports on the literature study that was conducted on frameworks. It discusses the definition and types of frameworks, and presents the methodology for the development of frameworks. It also presents the definition of a framework and the framework development methodology in the context of this research study. **Chapter** 3 presents the literature study that was conducted on data. It discusses the definition, types, taxonomy and sources of data. It also discusses the process of cleaning and preparing data for various cases. Furthermore, it presents the definition, types, applications and process of applying DA. In **Chapter** 4 the emphasis falls on ML algorithms. The definition, categorisation and application purposes of ML algorithms are presented. The chosen ML algorithms for this research study are presented in detail, including their methodologies, parameters, advantages and disadvantages. Different performance measurements of ML algorithms are discussed as well.

The developed decision support framework is presented in **Chapter 5**. A conceptual framework is introduced and further detailed using experiments and consulting applications in literature. Appendix A presents the datasets used in this research study. **Chapter 6** reports on the validation of the framework by presenting and synthesising the feedback of the SMEs and PEUs. Lastly, **Chapter 7** presents the research project summary and conclusions.

## 1.10 Conclusion: Chapter 1

In this chapter the research project was introduced. The problem background and motivation, problem statement, project objectives as well as the project methodology were discussed and a supportive, preliminary literature study was presented.

The following chapter will concentrate on the literature study performed on frameworks and framework development methodologies.

# Chapter 2

# Frameworks

The research project was introduced in the previous chapter. It stated the research problem and laid out the research rationale and motivation. The project scope and its objectives were introduced and the research methodology was developed. The research methodology stated that a literature study on frameworks and framework development is necessary to complete this research study.

This chapter will provide the definition of a framework, the distinction between theoretical and conceptual frameworks, and provide information about the different types of conceptual frameworks. Furthermore, it will discuss the definition of a framework within the context of this research study. Lastly, it will present the methodology chosen for developing the decision support framework for this research study.

## 2.1 Frameworks

The following section will describe the definition of a framework as stated in literature, the types of frameworks and further detail the types of conceptual frameworks. Lastly, the methodology for developing a framework, as set out by Jabareen (2009), is discussed.

### 2.1.1 The definition of a framework

A framework integrates existing theories, related concepts and empirical research for different research purposes. It is a model which is theoretically based and empirically supported for the purpose of conducting research and discussing the research related findings (Kumar & Antonenko, 2014).

Rocco & Plakhotnik (2009) identified the following five functions of frameworks:

1. Frameworks create a foundation for the study.
2. Frameworks illustrate how a study develops and progresses knowledge.
3. Frameworks enable the conceptualisation of the study.
4. Frameworks evaluate the research design, instrumentation, compilation and methods.
5. Frameworks provide a reference point for the analysis and understanding of the research findings.

### 2.1.2 Types of frameworks

Frameworks are mainly divided into two different types: theoretical and conceptual frameworks. *Theoretical frameworks*provide theoretical explanations while *conceptual frameworks* provide understanding of a study related experience (Jabareen, 2009).

1. Theoretical frameworks
   The purpose of a theoretical framework is to create a foundation for new theory development (Rocco & Plakhotnik, 2009). It enables the presentation and testing of a theory as well as the presentation of the associated experimental and conceptual work about the theory. A theoretical framework enables the investigation of a specific theory.
   Due to lack of theoretical framework literature, conceptual frameworks will be subsequently discussed.

2. Conceptual frameworks
   The purpose of a conceptual framework is to develop and systematise knowledge about associated concepts, issues or problems (Rocco & Plakhotnik, 2009). A conceptual framework consists of the theoretical and experimental work associated with the research problem or purpose, where the purpose is specifically not to advance a distinct theory. This framework creates a foundation of knowledge bases which are relevant to the research purpose. The conceptual framework provides a basis for the interpretation of the causal patterns or interconnections

across ideas, observations, events, concepts, knowledge and other aspects of experience (Svinicki, 2008).

The different types of conceptual frameworks will be discussed next.

### 2.1.3 Types of conceptual frameworks

Conceptual frameworks were classified into five different types by Shields (1998) and Shields & Tajalli (2006). Each classification is paired with a research purpose, research questions, research modes (methods or techniques) and statistics. A research purpose is the goal of the study and the reason why a study is conducted. Five different research purposes have been identified and are explained below in conjunction with the different types of conceptual frameworks.

1. Descriptive categories

   Descriptive category frameworks are linked to a descriptive research purpose which addresses the "what" question. This framework enables the analysis, description and classification of the objects under study. It assists with the identification of the characteristics of the objects and the determination of the shared characteristics to enable the grouping of objects (Shields & Tajalli, 2006).

2. Formal, causal hypotheses (if x then y)

   Formal, causal hypotheses frameworks are paired with an explanatory and prediction research purpose, which addresses the "why" question. This framework is used to identify the causes, to develop relationships of cause and effect, and to predict the characteristics of the effects.

3. Models of operations research

   Models of operations research frameworks are linked to a decision making and predictive research purpose. Models are complex formal hypotheses since they involve multiple hypotheses and variables. They are not limited to one objective or goal, and are used to predict the best or most efficient approach or decision (Shields, 1998).

4. Practical ideal type

   Practical ideal type frameworks are paired with an "understanding", gauging or analytical research purpose. The framework is used to help understand reality

and provide points of reference or standards. It uses the points of reference or standards to assess how close a situation is to the ideal or standard and to determine how the situation can be improved.

5. Working hypotheses
   Working hypotheses frameworks are linked to an exploratory or exploration research purpose since these frameworks enable and focus data and evidence collection. This type of framework facilitates discoveries and supports the advancement of investigations. Exploratory research is used when a problem, topic or issue is still in its early stages, and further conceptualisation and discoveries are still needed (Shields & Tajalli, 2006).

For this research project, a model of an operations research framework will be developed since it will be used for decision making purposes.

### 2.1.4 Developing a framework

The following process of developing a conceptual framework was designed by Jabareen (2009). It involves performing a literature study, identifying concepts in the literature and creating the framework based on these concepts and their relationships. Jabareen (2009) suggests building conceptual frameworks from existing multidisciplinary knowledge sources through a process of theorisation and utilising the grounded theory methodology. Jabareen's framework development methodology is summarised in Figure 2.1 and consists of the following phases:

1. Identify data sources
   A comprehensive set of data sources, including theoretical, empirical and practical data, should be identified and extensively reviewed. The data sources could include books, peer-reviewed and well-known scientific journals, articles and interviews. The sources should effectively represent the phenomenon under study and the practices which are related to the phenomenon. The data collection should enable holistic mapping to ensure validity.

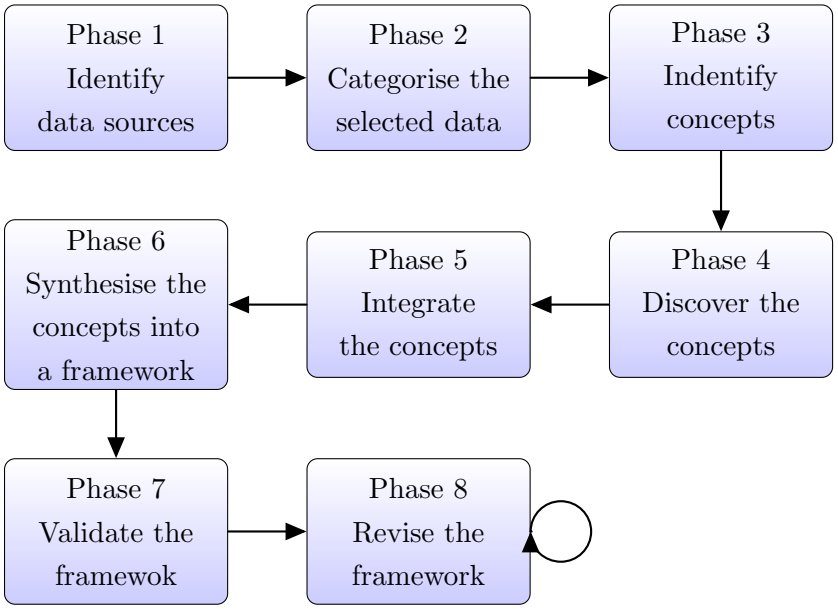


Figure 2.1: Jabareen's framework development methodology Jabareen (2009)

2. Categorise the selected data

   In this phase, the sources should be read and categorised. The categorisation can be based on the discipline or dimension, scale of importance and on the scale of representative power within each discipline.

3. Identify concepts

   By reading and re-reading the data sources, concepts are discovered and named. A concept is defined as follows: a concept consists of more than one component which defines the concept. Every concept has a history since every concept relates back to other concepts, and is created from other concepts.

4. Discover the concepts

   Each concept is deconstructed to identify its attributes, characteristics, assumptions, limitations, distinct perspectives and specific function within the conceptual framework.

5. Integrate the concepts

   Concepts with similarities can be grouped together to form a new concept. This is done to reduce the number of concepts to a manageable number of concepts.

6. Synthesise the concepts into a framework

   A framework is constructed by using the identified concepts. In this phase the researcher iteratively synthesises and resynthesises the framework until it "makes sense" in the light of the phenomenon. The researcher of the study should know how to build and recognise their conceptual framework given the research and literature review performed prior to the development of their framework.

7. Validate the framework

   In this phase, the developed framework should be validated. Validation determines whether the proposed framework is acceptable within the academic and practical environment.

8. Revise the framework

   Frameworks representing multidisciplinary phenomena are always dynamic and should be revised as new insights, literature and experiences become available.

## 2.2 The framework definition and framework development methodology for this research study

The following section will describe the definition of a framework from the perspective of this research study and detail the framework development methodology which will be implemented in this research study.

### 2.2.1 The definition of a framework for this research study

Given the literature review on frameworks as presented in the previous section and the preliminary literature review on the available frameworks for decision support regarding machine learning (ML) algorithms in both academic literature and practical implementations in Chapter 1, the researcher now presents her own definition of what a framework is for the purpose of this research study. This definition builds on the definition given in the previous section and is as follows:

A framework is a decision support tool which is *visually depicted* with *logical flows* and *directions* to assist the user in reaching a conclusion given his/her requirements. A framework aggregates the independent variables and presents the associated dependent variable(s). Furthermore, it indicates the most appropriate course of action given the independent variables of the application of the user.

People understand better when information is presented in a visual form. Visualisation provides a clear, condensed and summarised image of the information conveyed to the user. The logical flows in the forms of arrows indicating direction, easily lead the user through the image and avoids confusion since it is presented in a clear, rational, analytical and sequential order. The framework considers the independent variables and visually illustrates the conclusion of their interaction, *i.e.* the dependent variable(s). Details of the interaction is not presented as to not confuse the user and to reduce complexity by keeping the image simple. The interactions of the independent variables can lead to multiple dependent variables and multiple values per dependent variable. The dependent variables are evaluated in terms of their quantitative values. The framework indicates which dependent variable is most desired or most suitable for the user, based on the independent variables of the user and the values of the resulting dependent variables. The aim of the framework is to support the user in selecting the most beneficial dependent variable in a clear, easy to follow process.

### 2.2.2 Comparing the research methodology and the framework development methodology

The comparison of the Soft Systems Methodology (SSM) presented in Section 1.6.1 and Jabareen's framework development methodology is presented in Table 2.1. The SSM is a methodology used to address a problem using a general method whereas Jabareen's methodology focuses specifically on developing a framework. The SSM is the overhead research methodology in this research study and Jabareen's framework development methodology is followed for developing the decision support framework in this research study.

Table 2.1: Comparing the Soft Systems Methodology and Jabareen's framework development methodology

| **Soft Systems Methodology** | **Jabareen's framework development methodology** |
|---|---|
| 1. Consider the variation of the problem situation<br>2. Express the problem situation | 1. Identify data sources |
| 3. Define significant and purposeful root definitions | 2. Categorise the selected data<br>3. Identify concepts<br>4. Discover the concepts |
| 4. Construct conceptual models | 5. Integrate concepts<br>6. Synthesise the concepts into a framework |
| 5. Compare the conceptual model and the problem situation<br>6. Construct feasible and desirable changes<br>7. Define and implement action to improve the problem situation | 7. Validate the framework |
| | 8. Revise the framework |

### 2.2.3 The framework development methodology for this research study

In pursuit of the research objectives presented in Section 1.4, Jabareen's framework development methodology as presented in Section 2.1.4 is further detailed with the focus on developing a decision support framework for ML applications given the data characteristics and application purpose of the users' requirements.

1. Identify data sources

   The following set of data sources should be reviewed: data, definition of data, types of data, sources of data, data repositories, data analytics (DA), types of DA, processes of applying DA, DA purposes, ML, ML algorithms (including their strengths, weaknesses, pitfalls and limitations), types of ML applications, ML categorisation, ML performance measures, industrial engineering purposes and industrial engineering concepts focussing on value creation in businesses.

2. Categorise the selected data

   Possible disciplines could include: data, DA, ML, and industrial engineering for value creation in businesses.

3. Identify concepts
   Three main concepts are identified: data characteristics, application purposes and ML algorithms.

4. Discover the concepts
   Investigate the different data characteristics, application purposes and ML algorithms.

5. Integrate the concepts
   This might be an unnecessary step in the process, since the three main concepts have been identified in step 3.

6. Synthesise the concepts into a framework
   In the context of this work, the independent variables are the data characteristics and the application purposes, and the dependent variable is the ML algorithms. During this phase the relationships between the two independent variables and the dependent variable are determined, by performing literature studies, performing experiments and using applications found in literature. The findings are used to develop, improve and expand the framework.

7. Validate the framework
   The framework will be validated by consulting subject matter experts (SMEs) for their technical expertise and by consulting possible end-users (PEUs) to determine the interpretability and user-friendliness of the framework.

8. Revise the framework
   As new data types, application purposes and ML algorithms are developed, the framework should be revised and further developed to present the new insights to the users.

In this research study Phases 1 to 5 will iteratively occur since the exploration of each concept could lead to the discovery of new data sources and new concepts to investigate. Phase 8 of Jabareen's framework development methodology, revise the framework, is not applicable to this research study since the project concludes after the validation for the framework.

## 2.3 Conclusion: Chapter 2

In this chapter, the literature study concerning frameworks, types of frameworks and a framework development methodology were discussed. The type of framework to be developed in this research study was identified and the associated framework development methodology determined.

The following chapter will concentrate on the literature studies of relevant concepts, including data analytics, its applications and different processes of applying data analytics. Literature studies of data and data processing will be included as well.

# Chapter 3

# Data and data analytics

The previous chapter focused on frameworks and identified the framework to be developed in this research project. It presented the framework development methodology which will be followed in this research study.

The research in this chapter is based on a literature study of data analytics (DA) and the process of applying DA. Furthermore, this chapter will provide the necessary background to enable the process of applying DA by providing literature reviews on data, data preparation for DA, and the machine learning (ML) algorithms which will be employed by this research study.

## 3.1 Data analytics

As stated in Chapter 1, DA is the process of investigating and exploring data to derive insightful and relevant trends which can be used for a wide variety of applications, including decision support and process optimisation (Zhu *et al.*, 2014).

### 3.1.1 Types of data analytics

The focus of DA falls on the "extraction of actionable knowledge and insights from big data" (Rajaraman, 2016). There are four main types of DA, based on what was extracted and used from the data. The four types of DA is defined as follows:

1. Descriptive analytics

   Descriptive analytics presents the past in an easily understandable and interpretable format, given the data it used. The accumulated data is organised and

presented in the visual forms of charts, graphs, maps and diagrams for simplified visualisation. It enables the user to gain insight into what the data implies about the past.

2. Diagnostic analytics

   Diagnostic analytics identifies and determines unexpected relationships, trends and patterns among attributes or features in a large dataset. It is also known as exploratory or discovery analytics. Diagnostic analytics condenses raw data to smaller sets of information which are easier for humans to interpret compared to the large initial raw data. It enables users to make serendipitous discoveries and gain new insights from their data by quantitatively describing the main features of the dataset (Mujawar & Joshi, 2015; Rajaraman, 2016). These discoveries can lead to the development of countermeasures in cases with undesired outcomes.

3. Predictive analytics

   Predictive analytics uses the known data to infer what is most likely to happen in the near future. It enables forecasting, prediction and estimation. It examines the data (the past) to detect patterns and relationships between the inputs and outputs and then extrapolates these relationships forward in time to make predictions about the future (Mujawar & Joshi, 2015). The techniques employ time-series analysis since past sequential data is needed to discover patterns over time and it is used to make predictions with (Rajaraman, 2016). Predictive analysis can also be used to evaluate hypotheses.

4. Prescriptive analytics

   Prescriptive analytics identifies opportunities and indicates the applicable course of action to optimise solutions to existing problems. It identifies the most beneficial outcome and indicates what a user should do to achieve this goal (Rajaraman, 2016). The ultimate decision to follow this course of action still remains with the user. Prescriptive analytics is a type of predictive analytics with two additional components: actionable data and a feedback loop that monitors the consequence of the action taken. Additionally it can predict multiple futures where each is a possible course of action with an associated, most likely outcome (Mujawar & Joshi, 2015).

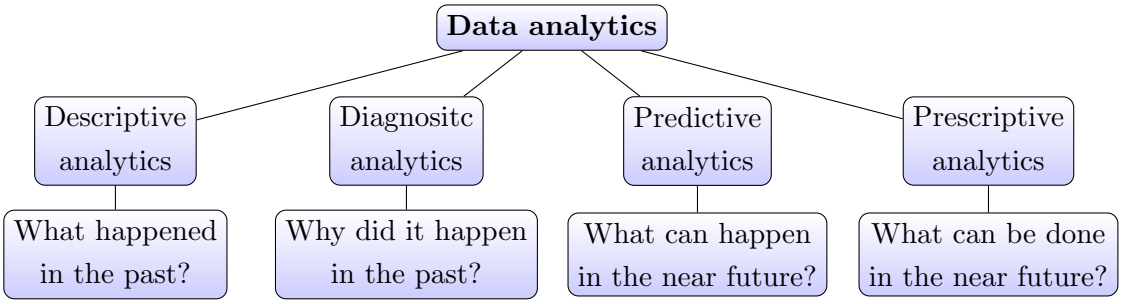


Figure 3.1: The types of data analytics (Mujawar & Joshi, 2015; Rajaraman, 2016)

### 3.1.2 The application purposes of data analytics

During the literature review conducted to publish Du Preez & Oosthuizen (2019b), Du Preez & Oosthuizen (2018), and Du Preez & Oosthuizen (2019a) different application purposes of DA were observed. The extensive review on DA applications performed by Köksal *et al.* (2011) contributes to these findings, as does the findings of Ngai *et al.* (2011), although this work is specialised in the financial fraud detection domain. The application purposes are identified and separated based on the type of knowledge gained. The purposes are subdivisions of the four types of DA, as illustrated in Figure 3.2. The purposes were subdivided based on the definitions of the four types of DA as presented in previous section. The application purposes of DA are not limited to the purposes presented here, for example, sequence discovery and time-series prediction are other possible purposes. The application purposes presented in this section will be used throughout the rest of this study.

The application purposes are as follows:

1. Association

   Association determines a set of "if-then" rules by identifying groups of items that frequently occur together and discovering relationships between them. The rules are learnt from the data (diagnostic analytics). It could be used for prediction (predictive analytics) as well.

2. Classification

   Classification groups similar instances together based on the groupings done in the past or using experience. It learns the trends in the data (diagnostic analytics) and use it to predict the classes of new data (predictive analytics).

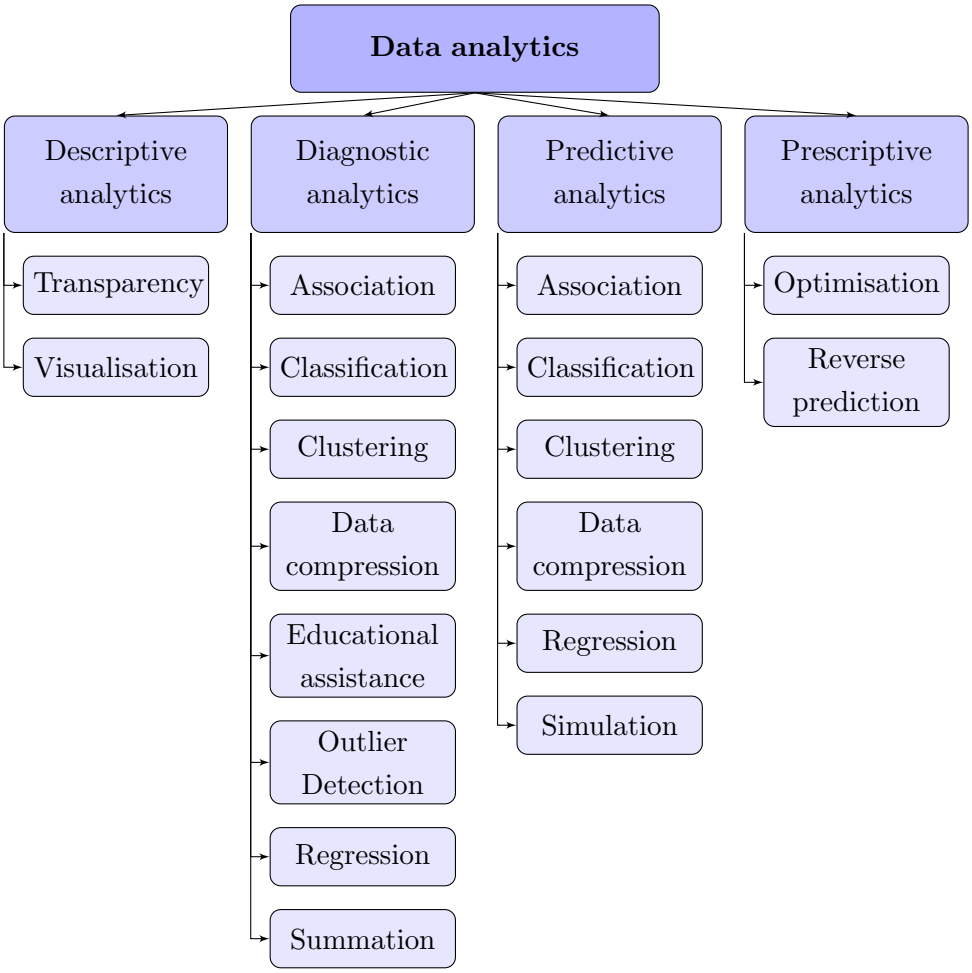


Figure 3.2: The application purposes of data analytics

3. Clustering

   Clustering groups similar instances together based on correlations, trends and patterns determined from the data. Unlike classification, the past experience of grouping is not available to assist the process. Given what was learnt from the data (diagnostic analytics), clustering can be applied as a method to predict the groups in new data (predictive analytics).

4. Data compression

   Data compression determines the most significant factors in the data or create new factors, which represent the data in fewer dimensions. It is also called 'data de-noising' (Marsland, 2014). The significant factors are learnt from the data

(diagnostic analytics) and could be used for prediction (predictive analytics) as well.

5. Educational assistance
   Educational assistance focus on gaining new insights and making new discoveries in the data. It can be used to train inexperienced employees (Nisbet *et al.*, 2009). Educational assistance is a subset of diagnostic analytics.

6. Optimisation
   Optimisation determines the values of the independent variables needed to yield the best or most effective dependent variables. The optimised solution provides a possible course of action. Optimisation is a subset of prescriptive analytics.

7. Outlier detection
   Outlier detection discovers anomalies, irregular patterns, anomalous behaviour or outliers in the data (Bose & Mahapatra, 2001). It learns what outliers are in the context of the data (diagnostic analytics).

8. Regression
   Regression determines the functional relationship between the independent and continuous-valued dependent variables. This can be used to see the effect and influence of the independent variables on the dependent variables (Ngai *et al.*, 2011). The relationships are learnt from the data (diagnostic analytics) and used for prediction (predictive analytics).

9. Reverse prediction
   Given a desired output, the developed model predicts the required input variables to achieve the desired outcome. This reverses the direction of prediction of the developed model. This is a subset of prescriptive analytics.

10. Simulations
    Simulations experiment with the developed model to create different "what-if" scenarios and determining the outcome of each (predictive analytics). It can lead to the development of countermeasures in cases with undesired outcomes. Additionally, it can be used to indicate a set of possible actions or solutions for the

user to choose. It can indicate desired course of action (prescriptive analytics), which can be regarded as optimisation.

11. Summation
    Summation discovers and interprets the trends and patterns found in the data. It presents the general characteristics which have been learnt from the data (Köksal *et al.*, 2011) and is a subset of diagnostic analytics.

12. Transparency
    Transparency gains insight as to what happened to the data during the DA venture. It provides clarity on the operations performed on the data before the final result of the DA was achieved. It is a subset of descriptive analytics.

13. Visualisation
    Visualisation illustrates the trends and patterns in the data visually. It enables the communication of complex patterns and relationships discovered in a DA venture in a clear, easily understandable presentation of data or functions. Visual characteristics such as colour, size and position can effectively be used to communicate the findings (Ngai *et al.*, 2011). It is a subset of descriptive analytics.

### 3.1.3 Data analytics, data mining and machine learning

As previously stated, *data analytics* (DA) is the process of investigating and exploring data to derive insightful and relevant relationships (Zhu *et al.*, 2014). It is a way of thinking and acting. It is not a tool, technique or technology (Prasad, 2016). It is the process of dividing a problem into simpler parts and using data to make inferences which are used for decision making and process optimisation. Typical DA tasks include descriptive, interactive, visual, diagnostic, predictive and prescriptive activities. Interactive activities enable the viewing of the data in terms of summarised statistical parameters. Visual activities enable the viewing of graphical displays to identify patterns or trends in the data (Nisbet *et al.*, 2009).

According to Nisbet *et al.* (2009), *data mining* (DM) is the application of ML techniques (but not limited to) to discover relationships and patterns in the data of large, noisy and messy datasets, which can lead to decisions and activities to increase benefit, for example, prediction, detection and diagnosis. DM includes the preprocessing

of data, application of modelling techniques and ML algorithms, interpretation of the results and the presentation of the mined information in a format which is useful for decision making (Alpaydin, 2010; Nisbet *et al.*, 2009). DM was designed to exploit large volumes of data (Brown & Brocklebank, 1997). DM combines methods and techniques, including visualisation, statistical analysis and induction to explore and model data. DM is not restricted to a single technique; it is an iterative process in which many tools, techniques and methods may be employed (Brown & Brocklebank, 1997). DM has various tasks, including clustering, rule discovery, classification and regression (Nisbet *et al.*, 2009). Each individual DM tool, technique, method or algorithm does not perform all the DA tasks. However, when combined they can perform more DA tasks. Thus, DM is a subset of DA.

As previously stated, *machine learning* which consists of algorithms (sets of rules) that employ mathematical and statistical techniques to give computers the ability to study, learn from, identify trends and patterns, and determine similarities in data. These algorithms automate the work of exploring the data and require only vague queries from the user (Moore *et al.*, 2009). As mentioned, ML algorithms infer their own rules from the data (experience), instead of following a set of step-by-step rules as in traditional programmed algorithms (Thurn & Anderson, 2017). It can be applied to different volumes of data and is not limited to large volumes of data. There are four main categories of ML algorithms: *unsupervised*, *semi-supervised*, *supervised* and *reinforcement learning*. The categories can be further detailed into tasks, including clustering, association rule discovery, classification and regression.

In summation, ML is a subset of DM and DM is a subset of DA. The relationship is illustrated in Figure 3.3. In general, DA is an overhead method of thinking and acting, while DM builds a model based on the patterns in the data. Data analytics utilises a greater variety of techniques and methods, for example, statistical methods, ML, visualisation methods, simple data fittings and more which could be both manually used in a step-by-step process or in an automated fashion. DM utilises a smaller variety of techniques and methods (a subset of the methods and techniques which DA employ), which focus on automating the exploration of large volumes of data (Nisbet *et al.*, 2009).

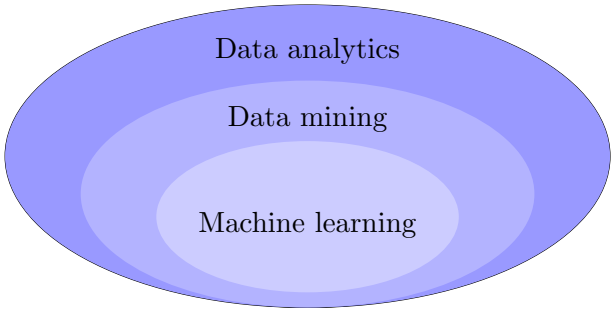


Figure 3.3: Data analytics, data mining and machine learning

### 3.1.4 The process of applying data analytics

A process consists of many steps or phases which are applied in a sequential order and it could be repeated in multiple iterations. A DA process is used to perform or enable DA. It is also called a framework from which to approach DA (Nisbet *et al.*, 2009). Two popular and widely recognised processes for applying DA are the CRoss-Industry Standard Process for Data Mining (CRISP-DM) and the Sample, Explore, Modify, Model and Access (SEMMA) process (Azevedo & Santos, 2008). They are called processes since they consist of a specific course of action followed to achieve a specific result (Azevedo & Santos, 2008).

#### 3.1.4.1 The CRoss-Industry Standard Process for Data Mining

The CRISP-DM was created by a consortium of European companies to serve as a standard process model for DA. It is an iterative framework which provides the industry standard for the implementation of ML by practitioners (Clark, 2018). The process consists of six major, sequential phases, each broken down into second-level generic tasks called *activities*, which can be further divided into specific tasks called *operations*. A fourth level of tasks called *process instances* sprout from the operations, which are domain specific, namely they address specific business problems with specific data (Nisbet *et al.*, 2009). The process is illustrated in Figure 3.4 (Azevedo & Santos, 2008). According to Mariscal *et al.* (2010), the CRISP-DM is a widely used DA process.

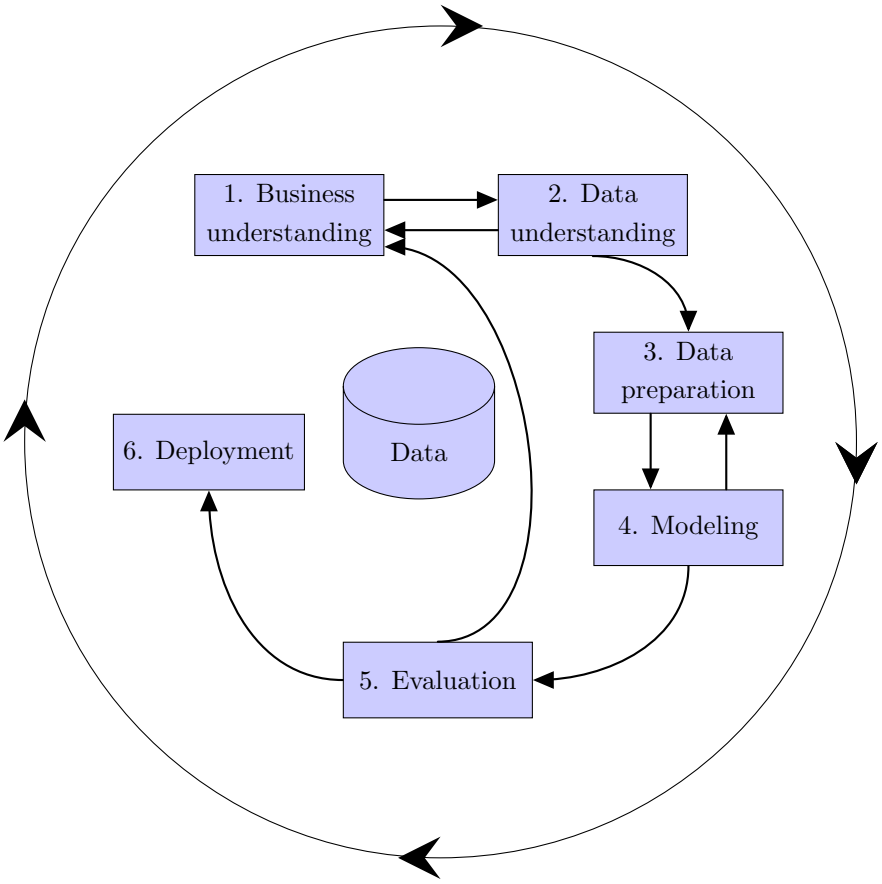


Figure 3.4: The CRoss-Industry Standard Process for Data Mining (Nisbet *et al.*, 2009)

The phases of the CRISP-DM are:

1. Business understanding

   The goal of the business understanding phase is to investigate and understand the objectives and requirements of the project from a business perspective. The objectives and requirements of both the business and DA are identified (Shafique & Qaiser, 2014). This knowledge is transformed into a problem description and project objectives, which are used to create a plan to achieve the project objective (Azevedo & Santos, 2008). It identifies success criteria, business terminologies and technical terms (Shafique & Qaiser, 2014). This phase consists of the following activities:

   (a) The business objectives of the project

   The goal is to identify the business need or problem and to investigate its

background. Stakeholders of the problem must be consulted to identify the effect of the problem and to create criteria of success for the solution to the problem. Next, the business objectives are formulated based on the success criteria. The benefits, drawbacks and the cost of the project should be provided to the stakeholders (Nisbet *et al.*, 2009).

(b) Assess the business environment

Inventory of the available resources must be taken, especially in the data modelling environment, for example, the DA tools and data support structure. The limitations must be determined, especially in the deployment environment. Plans should be made to acquire the relevant tools which are unavailable in the business environment and risks should be identified and addressed by creating contingency plans. This business assessment should be thoroughly documented with sufficient explanatory material and terminology for later reference. A domain expert can assist in the process and a business use case is developed (Clark, 2018).

(c) The data analytics project plan

The goals and objectives from the DA perspective are critical to the success of the project. It is based on the business objectives and ensures that a sufficient and appropriate model is developed and deployed for the problem (Niaksu, 2015). Firstly, the goals are identified and each of the goals are further detailed with objectives. The objectives are further broken down into tasks which will help implement the objectives. For each task a set of steps (subtasks), which need to be followed to accomplish the task, is identified. All the objectives with their associated tasks and steps are aggregated into a project plan. The project plan also details the start and end dates of each objective, task and step. It also identifies the resources required to complete the project plan (Nisbet *et al.*, 2009).

2. Data understanding

The data understanding phase focuses on understanding or familiarising oneself with the data by collecting the data and performing activities which lead to discovering insights from the data, identifying data quality problems or detecting subsets of data with interesting characteristics which can be used for hypotheses

formulation for hidden information (Azevedo & Santos, 2008; Shafique & Qaiser, 2014). The data understanding phase consists of the following activities (Nisbet *et al.*, 2009):

(a) Data acquisition

The data acquisition activity consists of three operations: data access, initial data collection and data integration. The various data sources should be identified and the applicable data for the modelling problem should be identified and extracted. Next, the data needs to be integrated by combining the different formats, aggregation levels and units to create a data map. The data map expresses how each data element of each dataset must be prepared to represent a common format and record structure.

(b) Data description

The data must be explored to determine its characteristics and it includes the following operations: determining variables, identifying cases or instances, employing descriptive statistics and producing a data description report. The data should also be examined to determine if it has an inherent scale to it (for example, it is given in Fahrenheit), if it contains biases or if the variables are possibly in conflict (Clark, 2018).

(c) Data quality assessment

The quality of data is measured according to the number of outliers, errors and missing values it contains. The data quality assessment activity consists of the following operations: identifying outliers, errors and missing values, as well as producing a data quality report (Nisbet *et al.*, 2009).

3. Data preparation

The data preparation phase includes all activities needed to create the final dataset from the initial raw dataset (Azevedo & Santos, 2008). The data has to be prepared by selecting, transforming and conditioning it to create a dataset with a suitable format for analytical modelling (Shafique & Qaiser, 2014). The underlying data structure should be such that it fits the input requirements of the statistical and DA algorithms. Data preparation activities include the following operations: data cleaning, transformation, imputation, weighting and balancing,

filtering, abstraction, reduction and derivation (Nisbet *et al.*, 2009). It is important to divide the data into training and validation datasets, respectively. The recommended ratio is 80/20, *i.e.* the training dataset consists of 80% of the original dataset and the validation dataset consists of 20% of the original dataset (Clark, 2018). Additional activities include sampling, normalisation and noise elimination (Niaksu, 2015).

4. Modelling

   The modelling phase focuses on selecting and applying various modelling techniques as well as calibrating their parameters to optimal values (Azevedo & Santos, 2008). Different models are built for the same problem (Shafique & Qaiser, 2014). The modelling activities include the following:

   (a) Select a modelling method

   The modelling method is selected by performing the following process instances:

   i. Select a modelling algorithm

   A modelling algorithm is chosen based on the outcome of the data understanding and data preparation phases. Further data preparation might be necessary to ensure that the input requirements of the modelling algorithm are met (Nisbet *et al.*, 2009).

   ii. Choose a modelling architecture

   The most suitable modelling architecture must be determined. Various architectures are available: a single model could be sufficient or multiple models, each with different parameters, are created and the one with the best performance is chosen as the final model. The structure or architecture of the same model can be adjusted with repeated experimentation to refine the model and to improve its performance (Clark, 2018). If the data slightly changes with time, a feedback process is needed in the model to iteratively adjust to the data and to improve its performance over time. A set of different algorithms can be developed or a model is developed on different samples of data and the results can be compared or combined to produce the final output. Consequently, it is required

that the most suitable architecture for the modelling problem and data is chosen (Nisbet *et al.*, 2009).

iii. Specify the assumptions made during the modelling process

Each modelling algorithm is based on underlying assumptions which should correlate with the data and the modelling goal.

(b) Develop an experimental design

To ensure a suitable experimental design, the response of the model under normal conditions must be determined and used as a control study to compare results of the model under various conditions.

(c) Build the model

The model is built by performing the following process instances:

i. Set the parameters

Some modelling algorithms are automatic and do not require parameters. Others are not automatic and default parameter values are used as a suggestion or baseline. These parameters should be investigated, since different datasets require moderately different model parameters for improved performance (Nisbet *et al.*, 2009). Multiple models with parameter values covering the whole available range in values should be build to ensure full optimisation; however, it is a computationally expensive task (Clark, 2018).

ii. Build various types of models

Multiple algorithms modelled on the same dataset will provide multiple perspectives of the data. It is recommended to develop multiple algorithms and then use a decision rule or heuristic to determine which output is the final output or how the output is combined to produce the final output.

(d) Assess the model

During the model assessment activity the generalisation of the model needs to be assessed. Generalisation is the ability to recognise similarities between different situations or data points, which can be applied in new situations or to new data points (Marsland, 2014). High generalisation enables the developed model to be reliable and resistant to variation or noise, which is

due to, for example, measurement inaccuracies. A model's generalisation is an indication of its appropriateness for application or deployment.

The recommended method to assess the generalisation of a model is by verifying it with an application in reality. However, this method is not always practical. Another method is to compare the model with the expectation of the output without a model. Various techniques to assist in model assessment are available, including tables, graphs and statistical measures of error. Typical metrics used for assessment include accuracy, precision, recall, reliability, effectiveness, sensitivity, specificity and functions, such as the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) (Niaksu, 2015).

A recommended practice is to use cross-validation, for example, k-fold cross-validation, which randomly splits the training data into smaller subsets and validates the model on random subsets of the data (Clark, 2018). This assessment process aids in developing models with improved performance and may also provide insight on where the model failed. Models are iteratively assessed until the assessments converge (Nisbet *et al.*, 2009).

5. Evaluation

   Both the developed model and the steps executed to develop the model are evaluated to ensure the model achieves the business and project objectives properly (Azevedo & Santos, 2008). After the outputs of the assessments converge, the model is presented as the final model. The results of the final model are evaluated in terms of accuracy. The modelling process is evaluated as a whole and the following steps for future modelling activities are determined. It is important to evaluate the accuracy and generalisability of the model as well as examine it for biases, including self-perpetuating bias. Futhermore, the model should be examined for unintended effects, for example, discrimination, which violate the principles of the business (Clark, 2018). The interpretation of the final model depends on the algorithm implemented to develop it (Shafique & Qaiser, 2014). The evaluation process also determines whether the business objectives have been properly achieved and the outcome of this phase is the approval or disapproval of the model for use in practical settings (Niaksu, 2015).

The following evaluation activities are reported in a modelling report (Nisbet *et al.*, 2009):

(a) Evaluate the model
The quality of the experimental design is reflected in the results of the model evaluation. There are various methods of evaluating the model, for example, the model can be applied to a new, unused dataset and assessment techniques can be used to evaluate its performance (Nisbet *et al.*, 2009).

(b) Evaluate the modelling process
The modelling goals, results and their correlation to the success criteria should be investigated. The successes and failures of the project must be identified.

(c) Future modelling goals
After evaluating the modelling goals, future modelling projects must be identified based on the merits of the present project. The modelling goals and approaches to accomplish them must be identified and a business case must be presented. The business case consists of the stakeholder support, business process, resource requirements and expected business benefits of these projects (Nisbet *et al.*, 2009).

(d) Produce a modelling report
Create a modelling report which summarises the evaluation results of the developed model.

6. Deployment
After the model has been created and evaluated, it has to be applied to achieve its goal (Azevedo & Santos, 2008). The obtained knowledge and results are organised, reported, presented and used when needed (Shafique & Qaiser, 2014). The model deployment phase consists of the following activities (Nisbet *et al.*, 2009):

(a) Plan the model deployment
Produce a model deployment plan.

(b) Plan model monitoring and maintenance
The following operations should be carried out: produce a model monitoring plan and a model maintenance plan.

(c) Final report
The following operations should be carried out: create a final modelling report and produce a final modelling presentation (Nisbet *et al.*, 2009).

(d) Review the project
Review and conclude the project.

This concludes the description of the CRISP-DM; the SEMMA process will be discussed next.

#### 3.1.4.2 The Sample, Explore, Modify, Model and Access process

The SEMMA process was developed by the SAS Institute, a leading company in business intelligence, and is incorporated into their knowledge discovery software platform, namely the SAS®Enterprise Miner (Mariscal *et al.*, 2010). It was developed to describe the process of conducting a DA project. It offers and enables the understanding, management, development and maintenance of DA projects (Shafique & Qaiser, 2014). The SEMMA process is illustrated in Figure 3.5. The stages of the SEMMA process are:

1. Sample
The initial stage of the SEMMA process is optional. During the sample stage, large sets of data are sampled from the initial dataset. These subsets must be big enough to consist of significant information and small enough to manipulate easily and quickly (Azevedo & Santos, 2008). SAS recommends using a sampling method which applies to a statistically representative and reliable sample of the initial dataset to ensure optimal cost and performances as well as reduced processing time to retrieve information from the data (Brown & Brocklebank, 1997). If a small niche which has an influence on the data is present, but is not well represented in a sample, SAS suggests using summary methods to discover the niche. Otherwise, general patterns will be traceable in a representative sample if it is present in the data (Brown & Brocklebank, 1997).


Figure 3.5: The Sample, Explore, Modify, Model and Access process (Mariscal *et al.*, 2010)

2. Explore

   The explore stage investigates and explores the data by searching and identifying relationships, patterns, trends, outliers, anomalies and unexpected trends in order to gain insight, understanding and ideas (Azevedo & Santos, 2008). It aids in refining the discovery process (Shafique & Qaiser, 2014). Both visual and numerical exploration can be utilised. If visual exploration does not indicate trends, statistical techniques (numerical exploration) can be used instead (Brown & Brocklebank, 1997).

3. Modify

   The modify stage focuses on modifying the data by assisting in the creation, selection and transformation of variables to direct the model selection process (Azevedo & Santos, 2008). This stage is also called the 'manipulate stage' (Nisbet *et al.*, 2009), since the data may need to be manipulated based on the discoveries of the exploration stage. Possible manipulations include the introduction of new variables, information, trends or groupings (Brown & Brocklebank, 1997). This stage may also search for outliers and reduce the dimension of the variables to reduce the dataset to the most significant data (Shafique & Qaiser, 2014). DM is an iterative, dynamic process and the DM models and methods can be updated when new data or information is available. Thus, when the mined data changes, the data needs to be modified.

4. Model

   The model stage enables the modelling of the data where software is allowed to automatically search for the optimal combination of data to reliably predict a desired outcome (Azevedo & Santos, 2008). Various modelling techniques are available, each with its own strengths, weaknesses and appropriateness for specific situations or datasets (Shafique & Qaiser, 2014).

5. Assess

   The assess stage evaluates the usefulness and reliability of the results of the DA process and predicts the performance quality of the developed model (Azevedo & Santos, 2008). There are two methods for assessing the model. A general method is to apply the model to a validation dataset, *i.e.* a subset of the initial dataset which is set aside before the sampling stage. A successful and useful model should work successfully on the validation dataset as it did on the samples or the data which was used to develop the model (Brown & Brocklebank, 1997). A second method is to apply the model on known data, where the true or correct output of the data is known and the results of the model on this data is evaluated against it.

#### 3.1.4.3 The data analytics process for this research study

Given the literature reviews on the CRISP-DM and the SEMMA process, the researcher decided to use the CRISP-DM throughout this research study for various reasons. These are:

- The CRISP-DM is more complete than the SEMMA process and provides clear guidelines (Shafique & Qaiser, 2014). The CRISP-DM is well documented and complete. All the phases are defined, organised and structured which enables a clear understanding and thorough revision of the project (Azevedo & Santos, 2008).
- The CRISP-DM accommodates business applications, not just academic applications (Nisbet *et al.*, 2009).
- The CRISP-DM is a widely used process (Mariscal *et al.*, 2010).
- It is difficult to use the SEMMA process since it is integrated into SAS tools while CRISP-DM is tool-agnostic.
- The CRISP-DM is closer to the real project concept than the SEMMA process (Mariscal *et al.*, 2010).

The CRISP-DM will be employed to fulfil Objectives 2, refitm:objective3 and 4 of the research objectives, by aiding in the design and execution of experiments to detail

the framework to be developed. For the remainder of this work, the study will only focus on the use of ML algorithms to limit the scope of methods and techniques which can be employed to perform DA.

Only Phases 3 and 4 are applicable to this research study. The first two phases of the CRISP-DM, 'business understanding' and 'data understanding' phases respectively, are not applicable to this study, since the goal of the study is not to evaluate a specific dataset(s) for the purpose of solving a specific business problem. The goal of the study is to provide a general guideline of choosing an ML algorithm, regardless of the industry from which the dataset originates or the problems which could be solved by applying ML to the dataset. Therefore, the last two phases of the CRISP-DM, the 'evaluation' and the 'deployment' phases respectively, are not applicable to this study. The developed models will only be evaluated in terms of the results they provide and not whether they achieve the business objectives of each dataset. The models will not be deployed, since the goal of the study is to create a conceptual decision support framework and not to use the developed models for future implementation.

## 3.2 Some characteristics of data

According to the English Oxford Living Dictionaries (2019), data is "facts and statistics collected together for reference or analysis". Tien (2013) described data as "values of qualitative or quantitative variables, belonging to a set of items". It is the "characters or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals". Data forms the basis of calculations from which information can be derived. Information can be used for reasoning and insight. Note that "data" is used in singular format throughout this thesis.

### 3.2.1 The taxonomy of data

Data may be categorised according to the types of their underlying variables. Data types are generally grouped into two categories, namely qualitative and quantitative data types. Data can represent various objects, including text documents, audio files, images, video files, time-series and transactional data (Rajaraman, 2016). Figure 3.6 illustrates the taxonomy of data, and this classification will be employed in this research study.

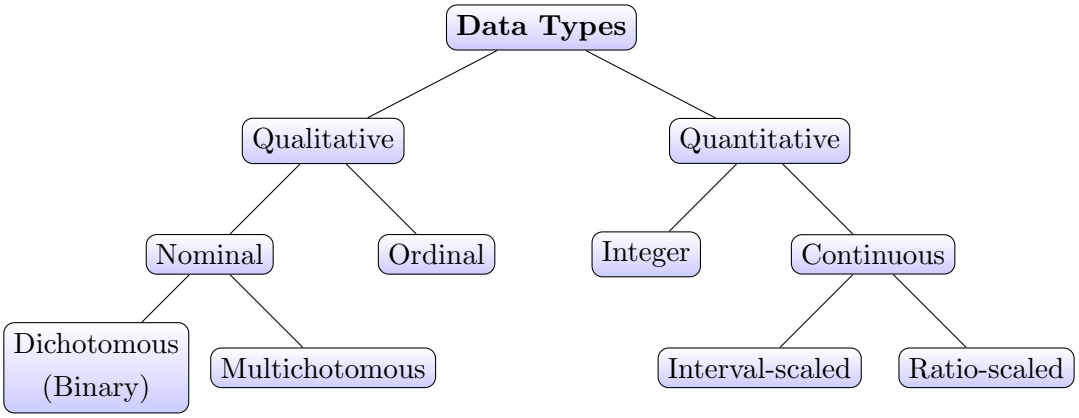


Figure 3.6: The taxonomy of data (Steynberg, 2016)

Data is either qualitative or quantitative. Qualitative data is subdivided into nominal and ordinal data, where nominal data is further divided into dichotomous and multichotomous data. Qualitative data is subdivided into integer and continuous data, where continuous data is further divided into interval-scaled and ratio-scaled data. It is important to note that all quantitative data is numeric; however, not all numeric data is quantitative.

*Qualitative data types* assume both textual and numeric values, for example, 'true' or '1'. The numeric values cannot be quantified, cannot be ranked and has no mathematical interpretation. It is inappropriate to perform mathematical operations on qualitative data (Hastie *et al.*, 2009). Qualitative variables are also known as factors, discrete variables or categorical variables, since they can be used to categorise data into groups or classes by providing labels. The enumeration of variables is possible where a numeric value represents a group or class which has a textual name, for example, '1' represents 'cat' and '0' represents 'dog'. Enumeration is a common practice in database design.

*Nominal data types* assume both textual and numeric values. There is no explicit ordering in the values where one value is bigger or more important than others (Hastie *et al.*, 2009). *Binary or dichotomous variables* can only take one of two possible values at a time, for example, 'true' or 'false', '1' or '0' (Bramer, 2007). *Multichotomous* variables may take more than two possible values at a time.

*Ordinal data types* are ordered categorical variables. There is intrinsic ranking or ordering between the values where one is more important than others; however, there is

no metric notion. This means that the difference between values $a$ and $b$ is not the same as the difference between values $b$ and $c$. Ordinal data types have meaning. Examples of ordinal variables are small, medium and large.

*Quantitative data types* are numeric in value and have meaning where one value or measurement is bigger than some but also smaller than other values. Measurements close in value are also close in nature (Hastie *et al.*, 2009).

*Integer variables* assume a value from a set of countable values or values which are isolated and separated by gaps. There is some sense of a metric notion where the difference between values 1 and 2 is the same as the difference between values 2 and 3; however, it is not the same as the difference between 1 and 3. Mathematical operations can be performed on integers in a meaningful way (Bramer, 2007).

*Continuous variables* may take on an infinite or uncountable number of values along a continuum (Montclair State University, 2017). There is a sense of a metric notion, the values have meaning and mathematical operations can be performed on them in a meaningful way. *Interval-scaled variables* are numerical valued variables which are measured at equal intervals from the origin or zero point, for example, the Celsius and Fahrenheit temperature scales. The origin does not necessarily represent the absence of the measured characteristic. There is no metric notion, only a measurement relative to the zero point. For example, 10 degrees is twice as far from the zero value as five degrees; however, saying 10 degrees is twice five degrees is meaningless. Also, when converting temperatures to an equivalent scale, the 'twice' relationship no longer applies (Bramer, 2007). *Ratio-scaled variables* are similar to interval-scaled variables; however, the origin or zero point does represent the absence of the measured characteristic. For example, Kelvin temperature, money and molecular weight. There is a metric notion where 10 dollars is twice five dollars (Bramer, 2007).

## 3.3 Data preparation and preprocessing

As stated previously, data can represent various objects (Rajaraman, 2016). For the remainder of this study, data types or data characteristics refer to the objects which data can represent and it is limited to the following: text, image, audio, video, time-series and transactional data.

The initial raw dataset has to be prepared by performing applicable selection, transformation and conditioning activities on it to create a dataset with a suitable format for analysis and analytical modelling (Sapp, 2017; Shafique & Qaiser, 2014). The data is prepared prior to the application of the DA model; thus, this stage is also called the 'preprocessing of the data' (Harrington, 2012). Data prepossessing activities include the following activities:

### 3.3.1 Data cleaning

Data is collected from a variety of sources and real-world data may contain missing and erroneous values due to subjective measurements, judgement errors and misuse or malfunctioning automatic recording equipment (Bramer, 2007). The goal of data cleaning is to improve the quality of the data, prior to the analysis thereof by detecting and removing both errors and inconsistencies. Increased data quality leads to increased reliability and validity. According to Nisbet *et al.* (2009), three primary objects have to be addressed in the data cleaning process: outliers, errors and missing values. Additionally, inappropriate variables or attributes should be removed from the dataset, since they decrease the accuracy and generalisation capability of the model developed thereon (Nisbet *et al.*, 2009).

#### 3.3.1.1 Outliers

An outlier is a single value or instance which falls outside the norm or the overall pattern (Moore *et al.*, 2009). It differs so much from the other instances that it prompts suspicion that it was created by a different or incorrect method (Nisbet *et al.*, 2009). Outliers are significantly different from the other instances in the dataset (Bramer, 2007). They are also called 'abnormal values', 'exceptions' or 'deviations'. There is no definitive way of identifying outliers since it is a matter for judgement.

Outliers are caused by a variety of sources, including unusual circumstances, equipment failure or they are genuine errors made during data recording and collection (Moore *et al.*, 2009). An outlier can be detected by using outlier detection algorithms. Generally, there are four types of outlier detection algorithms: based on critical distance measures, density measure, projection characteristics and data distribution characteristics (model-based grouping) (Nisbet *et al.*, 2009).

There are two methods to address outliers: remove or keep the outliers. The application or objective of the specific problem at hand determines which method to apply. Outliers are kept if the purpose is to detect abnormalities, unusual activities, learn different scenarios from the data or make major discoveries. On the contrary, outliers are discarded if the objective is to identify normal patterns or determine the typical response of a system since they reduce the predictability of the model (Nisbet *et al.*, 2009).

#### 3.3.1.2 Errors

Erroneous data is values which is incomplete, incorrect or captured in the wrong format for the dataset. Errors can originate from a variety of sources throughout the data collection and preparation process. It can be created by both human and device which create, capture, collect, transform, clean, filter, reduce, manage, sample, process and store data. Hellerstein (2008) identified the following four primary sources of data errors:

1. Data entry errors
   Data entry errors are errors due to human activity which can be related to typographic errors, misunderstandings of the data source or the invention of default values for the sake of convenience.

2. Measurement errors
   Measurement errors are errors due to human processes with errors in their design and execution. Possible sources include a human design of sensor deployment, incorrectly calibrated measurement devices and interferences from unintended or unanticipated signals.

3. Distillation errors
   Distillation errors are errors which originate from performing preprocessing, filtering, smoothing and aggregation on data. These processes influence the final analysis and can introduce errors in the distilled data.

4. Data integration errors
   Data integration errors are errors which occur due to the integration of data from multiple sources and the merging of pre-existing databases (Hellerstein, 2008).

It is difficult to ensure that erroneous data does not occur since the data collection and preparation process occurs across multiple sources, platforms, environments and organisations over potentially large spans of time and space. The following broad approaches to prevent the occurrence of erroneous data were presented by Hellerstein (2008):

1. Carefully design data entry interfaces
   Design data entry interfaces such that they prevent entries of non-existent data, are user-friendly, perform data type checks and provide limits on numeric values. These are called integrity constraints.

2. Use organisational management to improve data quality management
   Include the use of technological solutions, organisational structures and organisational incentives to improve total data quality management. Examples are utilising archiving and analysis processes, automating data capture and collection, capturing metadata to improve data investigation and interpretation, streamlining data collection and providing incentives for maintaining data quality.

3. Automate data auditing and cleaning
   Utilise existing computational techniques and automating technologies from both research and industry to prevent human interaction and error in data auditing and cleaning.

4. Perform investigative data analysis and cleaning
   Use operators with knowledge of the captured data to understand the characteristics of the dataset, to explore the dataset and to identify and rectify errors. Data cleaning tools and visualisation systems can be used to assist the operators (Hellerstein, 2008).

#### 3.3.1.3 Missing values

Missing values occur when data values are not recorded for all attributes or features.

Missing values can occur due to many reasons, including:

- Some attributes are not applicable to some situations or instances, for example, only male patients have a Y chromosome.
- The value should have been recorded; however, the measuring equipment was malfunctioning or misused.
- The value could not be obtained, for example, the patient to gain the data from, was unavailable.
- The data collection method was altered such that additional fields were added to the database after some data had been collected (Bramer, 2007).

There are various strategies available for addressing missing values. The following are three common strategies (Bramer, 2007):

1. Discard instances

   This strategy removes all instances containing at least one missing value from the dataset and use the remainder of the dataset. It is the simplest strategy; however, when instances are removed, the reliability of data analysis becomes questionable. In general, this method is discouraged and it is an inappropriate method when a high proportion of the dataset is removed. It is an applicable, conservative method when the proportion of missing values are small (Bramer, 2007). A similar method is to remove an attribute for which a high proportion of instances have missing values.

2. Determine the correct values

   This is the best preferred strategy; however, it is not always a practical solution. The data may no longer be available, may be inaccessible or in the case of an automated cleaning process, a manual entry is undesired. Approaches to achieve this strategy could be to implement procedures to detect the missing values, determine why they occur, improve the data collection process and consult subject matter experts (SMEs) to fill in the missing data (Fritchoff, 2010).

3. Imputing techniques

   This strategy uses the captured values of the instances or of an attribute to make a reasonable estimation of what the missing values should be replaced by. Imputation should be used with care. Through experimentation with different techniques, one can determine the appropriate technique for a particular dataset (Bramer, 2007). Several imputation techniques are available, including the following:

   (a) Mean imputation

   Mean imputation uses the average or mean of the attribute. This method is mostly used for continuous attributes.

   (b) Median imputation

   Mean imputation uses the median or most occurring value of the attribute. This method is mostly used for categorical attributes.

   (c) Associate rule imputation

   Association rule imputation determines a set of "if-then" rules which hold for instances in the dataset, given the attribute values (Bramer, 2007).

   (d) Hot deck imputation

   Hot deck imputation uses the value of that attribute which is prominent in similar data points (Marsland, 2014).

   (e) Maximum likelihood imputation

   Maximum likelihood imputation determines the function which describes the probability density map, based on the assumption that the predictor variables are independent. This function is used to determine the probability of a given missing value and maximises this probability (Nisbet *et al.*, 2009).

   (f) Other methods

   Other methods include single imputation, multiple imputation (Fritchoff, 2010), simple random imputation and multiple random imputation (Nisbet *et al.*, 2009).

### 3.3.2 Data transformation

Both numerical and categorical variables need to be transformed to an applicable format to increase modelling performance.

#### 3.3.2.1 Numerical variables

Numerical values of each feature of the data need to be standardised, especially if it has an inherent scale to it (Clark, 2018). *Standardisation* is the process of transforming all numerical values to a common range. Various standardisation methods are available. The most common method is to calculate the $z$-value or $z$-score which uses the mean and standard deviation of the data feature to calculate the new range (the transformed data), which has a mean of zero and a standard deviation of one. Another option is the zero-to-one standardisation which uses the maximum and minimum values of the data feature to transform the data to the range $[0, 1]$ (Nisbet *et al.*, 2009). Another option is to transform the values to the range $[-1, 1]$ (Marsland, 2014).

#### 3.3.2.2 Categorical variables

Textual categorical variables should be replaced by dummy variables (through the process of enumeration). A binary dummy variable ('0' or '1'), where '0' represents the absence and '1' represents the presence of a variable, is used. If a textual categorical variable has only two options, for example, 'true' and 'false', only one binary variable is required where '1' represents 'true' and '0' represents 'false'. In cases where the textual categorical variable has more than two options, the total number of binary dummy variables is equal to the number of options which occur in a textual categorical variable. An example is provided in Table 3.1. The disadvantage of this method is that the categorical variables have integer formats which might be treated as continuous values by the model instead of class labels. Additionally, dummy variables lead to information loss and reduce the ability to successfully apply the developed model on other new datasets (reduces the generality or generalisation capability of the model) (Nisbet *et al.*, 2009).

Table 3.1: Textual categorical variable transformation (Nisbet *et al.*, 2009)

| **Old** | **New** | | |
|---|---|---|---|
| Colour | Green | Yellow | Red |
| Green | 1 | 0 | 0 |
| Yellow | 0 | 1 | 0 |
| Red | 0 | 0 | 1 |

### 3.3.3 Normalisation

Weights are used to indicate the strength of the influence of one variable on another to determine the common or normal relationship between two or more variables. In the case of ML algorithms, the weights are learnt from the data in a case by case method instead of viewing all the data cases at once. Thus, the weights are incrementally and iteratively learnt (Nisbet *et al.*, 2009).

### 3.3.4 Filtering

Data filtering is the removal of instances or cases containing too much unnecessary or insignificant information. The goal is to reduce the noise and the variability in the data to enable the model to learn more accurately from the data. Filtering activities include low-pass, high-pass and time-series filtering. It is similar to signal processing where high frequency signal fluctuations are removed when they are either at the bottom or top of the range, or both. A low-pass filter accepts data which falls below a pre-specified level of acceptability and discards the rest of the data. Conversely a high-pass filter accepts data which falls above a pre-specified level of acceptability. A high pass filter accepts data which falls above a pre-specified level of acceptability and discards the rest of the data (Nisbet *et al.*, 2009).

### 3.3.5 Abstraction

Abstraction is the rearrangement of data to enable improved model development since the patterns are presented in a more recognisable way. It is similar to signal amplification. Abstraction activities include the following (Nisbet *et al.*, 2009):

1. Temporal abstraction
   Temporal abstraction maps variables over time to time stamps per variable instead.

2. Qualitative abstraction
   Qualitative abstraction maps numerical expressions to qualitative expressions.

3. Generalisation abstraction
   Generalisation abstraction maps an instance of an occurrence to its class, for example, grouping similar textual representations together.

4. Definitional abstraction
   Definitional abstraction maps a single data instance from one conceptual category to its counterpart in another conceptual category.

The first three abstractions are also called 'recoding' (Nisbet *et al.*, 2009).

### 3.3.6 Reduction

Data reduction includes three operations, namely data sampling, dimensionality reduction and value discretisation (Nisbet *et al.*, 2009).

#### 3.3.6.1 Data sampling

Data sampling reduces the number of instances or cases. The following four advantages, reasons and methods enable data sampling:

- It reduces the number of data instances used to develop the ML model, by using sampling methods like simple random sampling with the assumption that each sample instance has an equal probability of being chosen as does any other instance (Nisbet *et al.*, 2009).
- It assists in selecting data instances in which the output or response patterns are comparatively homogeneous, by performing partitioning on the dataset according to values of an attribute, for example, geographical location or eye colour. The partitioning should be on an attribute with limited options. It is advised to develop separate ML models on each partition or 'strata'. After partitioning, random sampling called stratified random sampling is performed in each partition.
- It balances the phenomena of rare events, since ML models are sensitive to unbalanced datasets. An unbalanced dataset is a dataset in which the occurrence of a single category of the output or target variable is relatively seldom compared to the rest. Dataset balancing includes two operations: oversampling rare categories or under-sampling common categories.
- It assists in the development of the ML model by providing datasets for training, validation and testing of the ML model through simple random sampling (Nisbet *et al.*, 2009).

#### 3.3.6.2 Dimensionality reduction techniques

Dimensionality reduction is the process of reducing inputs to lower dimensional representations thereof which still include the relevant information or features and reduces inapplicable information, including noise and outliers (Marsland, 2014). This process is performed by techniques which are called dimensionality reduction methods/techniques, which can be further divided into feature extraction and feature selection methods. It benefits ML algorithms greatly since it reduces the number of inputs, reduces the cost of extracting irrelevant information and makes the inputs easier to work with. It reduces the complexity of the algorithms by reducing the memory and computation requirements, increases the robustness of the algorithms, improves the results of the algorithms and makes the results easier to interpret and understand (Alpaydin, 2010; Harrington, 2012). Dimensionality reduction is used as a preprocessing step for ML applications (Marsland, 2014).

In feature selection methods a subset of relevant features is selected from the input and the rest is discarded. An example of such a method is subset selection. In feature extraction or feature derivation methods, a new set of inputs is created which contains fewer and new features which are derived from the original set of inputs by transforming and combining features (Alpaydin, 2010). Examples include both unsupervised and supervised learning techniques. In the context of unsupervised learning techniques, dimensionality reduction techniques include singular value decomposition (SVD), probabilistic graphical models (PGMs), correlation coefficients, the Gini index, principal components analysis (PCA), factor analysis (FA), multidimensional scaling (MDS), locally linear embedding (LLE) (Alpaydin, 2010), isometric feature mapping (Isomap), independent components analysis (ICA), the autoencoder (AE) (type of neural network) and the self-organising map (SOM) (Marsland, 2014; Nisbet *et al.*, 2009). Examples of supervised learning techniques include linear discriminant analysis (LDA) and multiple discriminant analysis. Clustering is another dimensionality reduction method, since similar data points are clustered together (Marsland, 2014).

#### 3.3.6.3 Value discretisation

Value discretisation is the conversion of continuous numeric variable values into discrete values. One method is to create bins, where the range of the numeric variable is divided

into sub-ranges of equal size and each individual numeric variable is replaced by the bin number of the bin into which it falls. The advantage of the binning process is that it reduces noise and variability in the data (Nisbet *et al.*, 2009).

### 3.3.7 Derivation

Derivation is the process of using the available dataset to derive other variables which are useful for the model development process. Derivation activities include the following (Nisbet *et al.*, 2009):

1. Target variable derivation
   This technique entails the determination of the target or output variable based on the input variables, for example, specifying a loss when the use rate drops below a threshold calculated from the data. The derivation is usually performed according to a heuristic or logical rule.

2. New variable derivation
   This method entails the calculation of a new variable based on the input variables, for example, calculating a rate by dividing one input variable by another in a meaningful way (Nisbet *et al.*, 2009).

3. Attribute-orientated derivation
   This technique entails the usage of a list of detailed categories of different attributes and combining specific values of those attributes to create a higher-level and more general expression for a new attribute through generalisation. An example is to classify workers (new attribute) according to their salary, number of owned vehicles and the value of their house. The opposite can be performed as well, where a new variable is created based the combinations of other attribute values to enable segmentation in the dataset (Nisbet *et al.*, 2009).

### 3.3.8 Data division

After the previous data preprocessing activities have been performed on the dataset, the final dataset is divided into specific subsets used during the different development stages of the model.

These subsets are:

1. Training dataset

   The training dataset consists of data instances which are randomly selected from the initial dataset to train the model (Nisbet *et al.*, 2009).

2. Validation dataset

   The validation dataset is a subset of the initial dataset which is set aside before the sampling and training stage. A successful and useful model should work as well on the validation dataset as on the samples or the data used to develop the model (Brown & Brocklebank, 1997). Thus, it is used to assess the predictability of the model after the model has been trained.

   The validation dataset enables further refinement or model enhancement since the results could be iteratively used to increase the performance of the trained model. The iterative process works as follows: the validation results of the current trained model are used to choose different parameters of the model and a new model is then trained on the training dataset with the new parameters. This trained model is then applied on the validation dataset and the results of the previous and current trained models are compared to determine which performed best. The process repeats until a trained model with desired performance has been developed. This process is called 'parameter tuning', since it is used to find the parameters such that the best model is developed.

3. Testing dataset

   In addition to the training and validation datasets, a testing dataset is utilised. The testing dataset is used to determine the final performance of the model after all modelling, training and development are done (Nisbet *et al.*, 2009). It enables an objective evaluation of the developed model since the previous two datasets were iteratively used to improve the learning and performance of the model. The testing dataset could be a subset of the original dataset or it could be collected separately from the original dataset.

## 3.4 Conclusion: Chapter 3

This chapter provided the literature study performed on data analytics, data and data preprocessing. Two processes for applying data analytics were discussed and the process of applying machine learning for data analytics were presented.

The following chapter will focus on the literature study on machine learning, the different types of machine learning and the different machine learning algorithms investigated for this research study.

# Chapter 4

# Machine learning

The previous chapter discussed literature on data analytics (DA), data and data preprocessing. The types of DA, DA application processes, types of data and data preparation techniques were presented in detail. The research methodology in Chapter 1 stated that a literature study on machine learning (ML) is necessary as well, to support this research study.

This chapter will provide a definition of ML and discuss various ML algorithms, including real-world applications, the algorithm methodologies and the advantages and disadvantages of the algorithms. Lastly, it will discuss the different methods of evaluating the performance of ML algorithms. Together, Chapters 3 and 4 fulfil Objective 1 of the research objectives.

## 4.1 Machine learning

As previously stated, *machine learning* algorithms are sets of rules that employ mathematical and statistical techniques to aid in the design of computer programs to enable the computer programs to independently study, identify trends and patterns, learn from and determine similarities in existing data (experience) without being explicitly programmed to do so (Sonosy *et al.*, 2016). Thereafter, predictions and decisions are made based on what was learnt and identified. In general, ML algorithms infer their own rules from the data instead of following a set of step-by-step rules as in traditional programmed algorithms (Thurn & Anderson, 2017).

Currently, ML applications are becoming increasingly popular due to the increasing availability of digitalised data, computational power and more powerful ML algorithms.

## 4.2 Types of machine learning algorithms

There are four types of ML algorithms, based on the data available and the associated type of learning developed. The terms labelled data and unlabelled data are important concepts in ML. *Labelled data* consists of both the input variables (called independent variables or features) and the associated output variables (called dependent variables, labels or targets) (Alpaydin, 2010; Marsland, 2014). *Unlabelled data* is data consisting only of the input variables or features and it does not contain any output variables or labels. The following four types of ML algorithms have been developed (Alpaydin, 2010; Marsland, 2014):

1. Supervised learning
   In supervised learning applications, labelled data is given as input to the ML algorithm and the goal is to learn the mapping or relationship which maps the input variables to the output variables. The model has to learn how to predict the output variables. The labels serve as a teacher or supervisor since they provide the desired level of output of the model and the model uses the labels to learn how to predict the output more accurately (Alpaydin, 2010).

2. Semi-supervised learning
   In semi-supervised learning applications, incomplete data (data consisting of both labelled data and unlabelled data) is given as input to the ML model. The goal of the ML model is to learn the mapping from the input variables to the output variables (Manning *et al.*, 2008).

3. Unsupervised learning
   In unsupervised learning applications, unlabelled data is given as input to the ML model and the goal is to learn similarities, regularities, patterns or structures in the data. The discovered patterns may be used for further DA (Alpaydin, 2010).

4. Reinforcement learning
   In reinforcement learning applications, complex labelled data is given as input to

the ML model. The input variables are sequences of actions and the labels are in the form of rewards and punishments. The goal of the ML model is to learn a strategy or policy to achieve the highest reward by learning from past sequences of actions and their associated rewards or punishments (Alpaydin, 2010). Reinforcement learning is also called 'learning with a critic' since a reward is given, while no correct strategy is provided. This type of learning falls between unsupervised and supervised learning (Marsland, 2014). Three important components of this type of learning are the ML model, the actions the model can perform and the environment the model interacts with (Portilla, 2017).

## 4.3 Classes of machine learning algorithms

The results of the ML algorithms enable the generation of knowledge which could be used for prediction, prescription, decision support and optimisation objectives (Paschek *et al.*, 2017). In general, ML algorithms are divided into six classes, based on the desired output of the ML model. Figure 4.1 illustrates the relationship between the six classes and the four types of learning. To enable the class of ML, the associated learning is required. The classes are similar to the application purposes listed in Subsection 3.1.2; however, in this context each of the six classes was further investigated by data analysts and engineers. Specific ML algorithms were designed and developed to achieve these specific purposes.

The six different classes of ML algorithms are as follows:

1. Association algorithms

   The goal is to find interesting relationships between specific values of categorical variables in large datasets. Variable types include simple categorical, multiple target and/or dichotomous variables. The relationships can be represented in two forms: frequent item sets or association rules. Frequent item sets are sets of items that frequently occur together. An example of an item set is {bread, milk} (Harrington, 2012; Nisbet *et al.*, 2009). Association rules are the "if-then" rules which are created from the frequent item sets if there is a strong relationship between items in the item set. The "if-then" rules are used to explain the relationship of the items in the item sets (Köksal *et al.*, 2011). An example of an association rule is {bread} $\rightarrow$ {milk}.

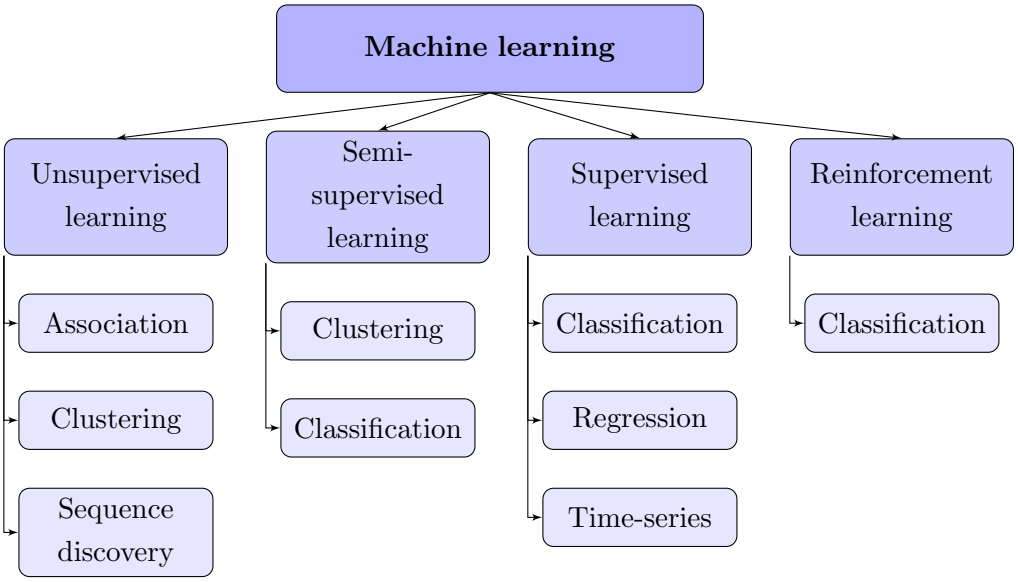


Figure 4.1: The relationship between the four types of learning and the six classes of machine learning

Association identifies groups of items which frequently occur together, by performing two sequential steps: find the frequent item sets and generate the association rules. Association algorithms employ unsupervised learning since hidden patterns or regularities are uncovered (Nisbet *et al.*, 2009). Association rule mining (ARM) is the process of extracting association rules (Bramer, 2007). It is also called 'market/shopping basket analysis', 'association analysis' and 'association rule learning' (Alpaydin, 2010; Nisbet *et al.*, 2009). Applications include the retail industry, website traffic analysis and medicine. Examples of association algorithms include the Apriori algorithm, Eclat algorithm and FP-Growth (frequent pattern growth) algorithm.

2. Sequence algorithms

   Sequence algorithms are similar to association algorithms; however, they determine sequential or temporal associations in the data since they are concerned with the order in which a group of items occur (Nisbet *et al.*, 2009). Sequence algorithms employ unsupervised learning as well. The application of sequence algorithms is also called 'sequence pattern mining', 'pattern discovery' or 'pattern analysis'. Applications include shopping patterns, DNA sequencing, web

log streams and linguistic patterns. Examples of sequence algorithms include Apriori-SPM (based on the Apriori algorithm), GSP algorithm, Sequential PAttern Discovery using Equivalence classes (SPADE), FreeSpan, PrefixSpan and MAPres.

3. Clustering algorithms

   Clustering algorithms use the features or attributes of each individual instance to determine patterns in or characteristics of the data (entire group of instances). The algorithm then groups individual instances into measurably similar groups called *clusters* based on the discovered similarities, patterns or characteristics (Guillén *et al.*, 2010). The instances in a cluster are similar to the other instances in the same cluster, but dissimilar to the instances of the other clusters (Ngai *et al.*, 2011; Nisbet *et al.*, 2009). There are two types of clustering: hard clustering and soft clustering. In hard clustering an instance may only belong to one cluster, while in soft clustering an instance's assignment is represented as a distribution over all the clusters (Manning *et al.*, 2008).

   Clustering can be performed according to various measures, including distance measures, density measures and distribution measures (Manning *et al.*, 2008; Nisbet *et al.*, 2009). Table 4.1 provides a summary of different clustering algorithms and the measurement types they are based on. Clustering can be used for cluster identification, anomaly detection and data compression (also called dimensionality reduction). There are both external and internal criteria for measuring clustering quality. External criteria evaluate the clustering results based on a dataset which is different from the one used to perform the clustering. Internal criteria evaluate the clustering result based on the dataset that was used for the clustering process itself.

   Clustering algorithms are unsupervised learning algorithms, since unlabelled data is used for clustering. They are also called 'data segmentation' or 'data partitioning' methods. Applications include customer segmentation, document clustering and image compression. Examples of clustering algorithms include one-class support vector machines (one class SVMs), $k$-means clustering (KMC), fuzzy $k$-means clustering and $k$-medoids clustering.

Table 4.1: A summary of types of clustering algorithms (Bijural, 2013; Moin & Ahmed, 2012)

| **Measure type** | **Types of clustering** | **Examples** |
|---|---|---|
| Distance | Partitioning methods (Centroid-based algorithms) | KMC, $k$-medoids clustering, $k$-medians clustering, fuzzy KMC, MS |
| | Hierarchical methods (Connectivity-based algorithms) | Agglomerative approach (bottom-up) Divisive approach (top-down) |
| Density | Density-based methods | DBSCAN, CLARANS, OPTICS |
| Distribution | Model-based methods | Gaussian mixture models, STING, CLIQUE, Expectation-maximisation |

Hierarchical clustering (connectivity model), expectation maximisation (EM), (Gaussian) mixture models (GMM) with expectation maximisation, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), CLARANS, OPTICS, STING, CLIQUE and mean shift clustering (MS) are examples of clustering algorithms as well (Alpaydin, 2010).

4. Classification algorithms

   Classification algorithms perform prediction by building a model based on labelled data and using it to predict the categorical labels or *classes* of new unknown instances (Harrington, 2012). Classification problems are discrete, since each instance belongs only to one class and the entire set of classes covers the entire possible input space (Marsland, 2014). The task is to learn the mapping from the input to the nominal or categorical output and the resulting model can be presented as classification rules or formulas (Alpaydin, 2010). Classification algorithms perform supervised learning (Marsland, 2014) since they aim to replicate a categorical distinction that has been given (labels) (Manning *et al.*, 2008).

   Classification can be performed according to various measures, including distance measures and density measures. Examples of classification algorithms include some neural networks (NNs) (including multilayer perceptrons (MLPs) and radial basis function NN (RBFNN), restricted Boltzmann machines (RBM) (Marsland, 2014), linear discriminant analysis (LDA), decision trees (DTs), classification and regression trees (CART), AdaBoost, Naïve Bayes or Baysian classifier (NB) (Alpaydin, 2010), logistic regression (LogReg) and the ridge classifier.

Table 4.2: The three different output options of classification algorithms (Bijural, 2013)

| **Output type** | **Definition** | **Applicable algorithms** |
|---|---|---|
| Binary | An instance could belong to one of two classes | NB, DT, RF, KNN, LDA, QDA, GPC, SVC, LogReg, MLP and ridge classifier |
| Multi-class | An instance can belong to one of many classes; however, only to one class at a time | |
| Multi-label | An instance can belong to many classes at the same time | DT, RF, KNN and ridge classifiers |

Random forests (RFs), $k$-nearest neighbours classifiers (KNNs), some support vector machines (SVMs) (including support vector classification (SVC)), quadratic discriminant analysis (QDA) and the Gaussian process classifier (GPC) are examples of classification algorithms as well. Table 4.2 lists the three possible output options of classification algorithms and the classification algorithms suited for each output (Pedregosa *et al.*, 2011).

5. Regression algorithms

   Regression algorithms apply statistical and mathematical processes to identify trends, patterns or features in data. They identify the most significant relationships and indicate the strength of the impact of independent variables on dependent variables. Regression algorithms perform supervised learning since the task is to learn the mapping from the input to the continuous output (labels) and the resulting model can be presented as a numeric function (Alpaydin, 2010). The relationships are used for various reasons, including predictions (Sharma *et al.*, 2017), association identification, time-series modelling and to determine the most influential factors in a dataset. Association identification determines rules which govern the relationships between variables and/or groups of variables (Bose & Mahapatra, 2001). The most influential factors can be utilised to improve prediction accuracy (Sun *et al.*, 2017).

   Regression predicts a continuous output variable by using the input/training data, whereas classification groups data into classes; thus, the outcome is categorical discrete values. Classification problems can be turned into regression problems by using an indicator variable which indicates the class of an instance. The regression algorithm is then trained to predict this indicator variable. Another option is to

perform repeated regression, one repetition per class, where the indicator variable has a value of 1 for instances belonging to the utilised class and 0 for the rest of the instances (Marsland, 2014).

Examples of regression algorithms include Gaussian process regression or regressor (GPR), regression-based KNN or $k$-neighbours regressor (KNR), linear regression (LinReg), multivariate linear regression (MLR), real AdaBoost or AdaBoost-R, DTs, CARTs, RFs, NNs and support vector regression (SVR) (Jiménez *et al.*, 2017; Lee *et al.*, 2014; Lin *et al.*, 2017; Šajn & Kukar, 2011).

For regression output there are two possible options: single output and multi-output regression. In single output regression only one target variable is predicted per instance. In multi-output regression, more than one target variable is predicted per instance (Pedregosa *et al.*, 2011).

6. Time-series algorithms

   Time-series algorithms are concerned with the order in which data occurs and they determine the sequential or temporal associations in the data with which to make continuous valued predictions. They are similar to regression algorithms and employ supervised learning as well. The relationship between time-series and regression algorithms is similar to the relationship between sequence and association algorithms; however, sequence and association algorithms are concerned with groups of items (clusters) whereas time-series and regression algorithms are concerned with continuous valued output.

   Examples of time-series algorithms include hidden Markov-models (HMMs), Gaussian processes, multilayered NNs and LinReg. Some regression algorithms can be used where the time-series data is altered to accommodate the requirements of the regression algorithms while still reflecting the time characteristic.

## 4.4 The selected machine learning algorithms

For this research study, the selected ML algorithms belong to the clustering, classification and regression algorithm classes since these classes of algorithms are the most frequently used in literature and practice. The selected algorithms are subsequently described in more detail, with summaries of their essential characteristics.

The selected algorithms for this study will now be presented in three classes, namely clustering, classification and regression, using the following structure:

1. Applications of each algorithm in the class.
2. Brief descriptions of each algorithm, as well as pointers to literature for essential detail (the researcher does not deem it necessary to present the detail in this document).
3. Advantages and disadvantages of the algorithms in the class.
4. Presentation of some performance metrics of the algorithms in a given class.

### 4.4.1 Clustering algorithms

The following unsupervised clustering algorithms will be covered by the study: agglomerative hierarchical clustering (AHC), DBCAN, KMC, mini-batch KMC (miniKMC), MS and the one-class SVM. These algorithms are most popularly used in practice and are well supported in *Python*. Table 4.3 presents the clustering algorithms with the following characteristics for each: 1) the source where they can be researched in detail, including their mathematical formulations, 2) applications found in literature for a variety of data types, including textual data, image data, video data, audio data, time-series data and transactional data, and 3) the application purposes as mentioned in Subsection 3.1.2.

#### 4.4.1.1 Agglomerative hierarchical clustering

The goal of hierarchical clustering is finding groups of instances such that instances in a group are more similar compared to instances in other groups (Alpaydin, 2010). Hierarchical clustering algorithms build nested clusters by merging or splitting repeatedly based on similarity or dissimilarity respectively. (Pedregosa *et al.*, 2011). In order to determine the similarity, a measurement which is based on the distances between instances is used.

Initially, AHC creates $N$ groups, where $N$ is the number of instances; thus, each group contains one instance. Next, similar groups are repeatedly merged to form larger groups until one group is formed. Thus, it moves from the bottom upwards.

Table 4.3: A summary of the applications of clustering algorithms

| Algorithm | Source | Data Types | Application purposes |
|---|---|---|---|
| All algorithms | | | Clustering<br>Visualisation |
| AHC | (Alpaydin, 2010)<br>(Pedregosa *et al.*, 2011) | Textual data (Cimiano *et al.*, 2004; Lin *et al.*, 2014)<br>Image data (J. & C., 2009)<br>Audio data (Pellegrini *et al.*, 2009)<br>Transactional data (Hung & Tsai, 2008)<br>Time-series data (Rani & Sikka, 2012) | Educational assistance |
| DBSCAN | (Pedregosa *et al.*, 2011) | Textual data (Yua & Jin-Sheng, 2011)<br>Image data (Bandyopadhyay & Paul, 2013)<br>Video data (Bewley *et al.*, 2014; Ranjith *et al.*, 2015)<br>Audio data (Mass *et al.*, 2014)<br>Transactional data (Backlund *et al.*, 2011)<br>Time-series data (Rani & Sikka, 2012) | Outlier detection |
| KMC<br><br>MiniKMC | (Alpaydin, 2010)<br>(Marsland, 2014)<br>(Harrington, 2012)<br>(Manning *et al.*, 2008) | Textual data (Lin *et al.*, 2014)<br>Image data (Ray & H. Turi, 1999)<br>Video data (Batra *et al.*, 2008)<br>Audio data (Tsunoo *et al.*, 2009)<br>Transactional data (Ghorbani & Farzai, 2018)<br>Time-series data (Das *et al.*, 1998; Rani & Sikka, 2012) | Data compression<br>Educational assistance<br>Outlier detection<br>Simulation<br>Summation |
| MS | (Pedregosa *et al.*, 2011) | Image data (Carreira-Perpiñán, 2015)<br>Video data (Comaniciu *et al.*, 2000) | Data compression<br>Educational assistance<br>Outlier detection<br>Simulation<br>Summation |
| One-class SVM | (Pedregosa *et al.*, 2011)<br>(Alpaydin, 2010)<br>(Schölkopf *et al.*, 2001) | Textual data (Shravan Kumar & Vadlamani, 2017)<br>Image data (Pla *et al.*, 2013)<br>Video data (Junejo *et al.*, 2010)<br>Audio data (Omid Sadjadi *et al.*, 2007)<br>Time-series data - pre-process data (Ma & Perkins, 2003)<br>Transactional data (Lamrini *et al.*, 2018) | Data compression<br>Educational assistance<br>Outlier detection<br>Simulation<br>Summation |

Divisive clustering achieves the same result as AHC; however, it approaches the dataset from the opposite direction. Initially, a group containing all the training instances is created and thereafter the group is divided into smaller groups which are repeatedly divided until each group contains a singular instance.

The AHC algorithm merges two similar groups based on a minimum distance which consists of two factors: the distance measurement and the linkage criterion. Various types of distance measurements are available, including Euclidean, Minkowski and city-block. The linkage criterion specifies which distance between groups is used for the clustering decision. Various linkage criteria are available, including single-link, complete-link (maximum-link), average-link, centroid-link and ward linkage. Single-link clustering is defined as the shortest distance between all possible pairs of instances in the two groups. Average-link clustering is defined as the average distance between all possible pairs of instances in the two groups. Centroid-link clustering is defined as the distance between the centres or centroids (means) of the two groups. Ward clustering determines the sum of squared differences between all possible pairs of instances in the two groups. The mathematical formulae are available in Alpaydin (2010).

The developed AHC model can be visualised by using a dendrogram. The dendrogram is a hierarchical tree-like structure where leaves represent instances which are grouped in the order in which they were merged (Alpaydin, 2010). It represents the hierarchy of clusters (Pedregosa *et al.*, 2011). The dendrogram may be intersected at any level to retrieve the required number of groups. An example of a dendrogram is provided in Figure 4.2, where there is a clear division between the two classes. A small height (measured on the vertical axis) before joining two groups indicates that the groups are quite similar and *vice versa* for larger heights.

#### 4.4.1.2 Density-Based Spatial Clustering of Applications with Noise

The DBSCAN algorithm has a generic view on clusters. Clusters are high-density areas separated by low-density areas. The DBSCAN algorithm focuses on core instances, which are instances in high-density areas. Density is measured by two possible values: the distance between neighbouring instances (*eps*) and the number of neighbouring instances ($min_{\text{instances}}$).

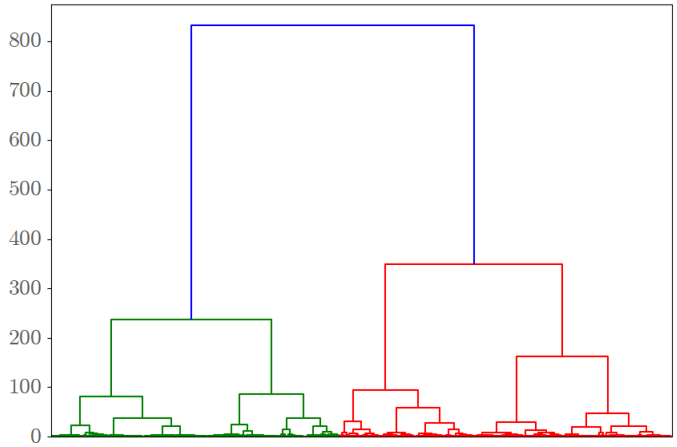


Figure 4.2: An example of a dendrogram of agglomerative hierarchical clustering with ward linkage which was performed on audio data of cats and dogs. The instances are on the horizontal axis and the height or depth on the vertical axis.

Formally, a *core instance* is an instance in the dataset which is surrounded by a number of neighbouring instances equal to or more than the $min_{\text{instances}}$ within a radius of *eps* around the core instance itself. The higher the $min_{\text{instances}}$ or the lower the *eps*, the higher the density (Pedregosa *et al.*, 2011).

A cluster is found by repeating the following process: find a core instance, determine all neighbours which are also core instances, determine which neighbours of the neighbours are core instances, and so forth. The non-core instances in the cluster are non-core neighbours of the core instances. They are located on the fringes of the cluster. Outliers are defined as non-core instances which are a minimum distance of *eps* away from any core instance (Pedregosa *et al.*, 2011). An example of DBSCAN is illustrated in Figure 4.3, where the larger coloured dots represent the core instances, the smaller coloured dots represent the non-core instances and the black dots represent the outliers. Each colour, excluding the black outliers, indicates a different cluster.

#### 4.4.1.3 $k$-means clustering

The KMC or $k$-means algorithm clusters the data into $k$ different clusters or groups which are represented by their centre points or centroids. It is called $k$-means clustering since $k$ unique clusters are created and each cluster centre is the mean or average of the data points in that particular cluster.

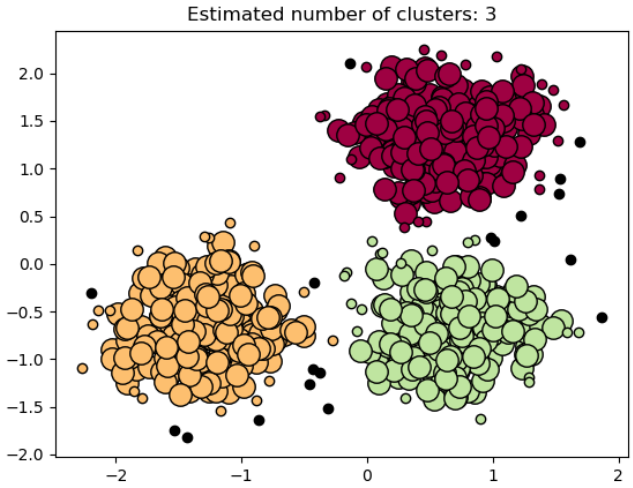


Figure 4.3: An example of the clustering results of Density-Based Spatial Clustering of Applications with Noise (Pedregosa *et al.*, 2011)

The KMC algorithm performs clustering by iteratively assigning each data point to its nearest cluster (based on minimising the distance to the centre) and moving the centre points based on the mean of the data points in each cluster until all the distances from the data points in each cluster to their centre point are minimised (Marsland, 2014). The algorithm calculates the distances within the Euclidean space (Marsland, 2014).

The KMC algorithm is a hard-clustering algorithm. It can be used to determine the distance from each data point to each cluster centre, to assign data points to clusters (by choosing the cluster with minimised distance between its centre point and the new data point) and to remove noise from data by replacing data points with their cluster centres (Marsland, 2014). Therefore, it is a partitioning and centroid-based clustering technique.

Variations of the algorithm include bisecting KMC, $k$-medoids clustering and miniKMC. More information regarding the algorithm methodology, algorithm variations, initialisation methods and terminating conditions can be found in Alpaydin (2010), Marsland (2014), Pedregosa *et al.* (2011), Manning *et al.* (2008) and Harrington (2012).

#### 4.4.1.4 Mean shift clustering

The goal of MS is to determine groups of high density instances in a smooth density of instances. It identifies regions and calculates the centroids of each region as the mean of the instances within those regions. It then iteratively moves towards regions with higher density instances and determines the centroids of the new regions. The algorithm terminates when the changes in the centroids are negligible, *i.e.* when areas of higher densities are not found (Pedregosa *et al.*, 2011). As a post-processing stage, the centroid candidates are then filtered to remove near duplicates and to keep centroids in higher density regions. The remaining set of centroids is the final set of centroids (Pedregosa *et al.*, 2011). The MS algorithm is a centroid-based clustering technique since it uses the centroids to perform clustering. The region size is determined by the user and determines the number of centroids. The mathematical formulae of MS are available in Alpaydin (2010).

#### 4.4.1.5 One-class support vector machine

The one-class SVM is presented in the following section, since it is a variant of SVMs. Originally SVMs were designed for classification; however, they were adapted to create a variant suitable for clustering.

#### 4.4.1.6 The advantages and disadvantages of the selected clustering algorithms

Table 4.4 presents the advantages and disadvantages of each clustering algorithm. The one-class SVM is not included since its advantages and disadvantages are presented together with the advantages and disadvantages of SVMs in the following section.

Table 4.4: Advantages and disadvantages of the selected clustering algorithms

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **AHC** | 1. It can handle large input datasets.<br><br>2. Single-link clustering is computationally efficient and it provides better performance with larger datasets. | 1. It is computationally expensive.<br><br>2. Ward clustering is limited in terms of distance metrics, since it does not support non-Euclidean distances. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **AHC** (cont.) | 3. Average-linkage clustering performs well with non-Euclidean distances (Pedregosa *et al.*, 2011).<br><br>4. Ward clustering provides the most uniform cluster sizes. | 3. Single-link clustering may lead to uneven cluster sizes and is not robust to noisy data.<br><br>4. The trained model cannot be stored. |
| **DBSCAN** | 1. It has a minimal number of input parameters.<br><br>2. It performs well when separating high-density clusters from low densities.<br><br>3. It discovers clusters with arbitrary shapes.<br><br>4. It identifies outliers. | 1. It performs poorly with high dimensional data.<br><br>2. It performs poorly when separating clusters with similar densities (Pedregosa *et al.*, 2011).<br><br>3. The trained model cannot be stored. |
| **KMC** | 1. It is efficient (Manning *et al.*, 2008).<br><br>2. It is easy to implement and simplistic in nature (Harrington, 2012).<br><br>3. It mostly converges to a local minimum of the error function.<br><br>4. The trained model can be stored. | 1. It is a local search procedure.<br><br>2. It is sensitive to initialisation and local minima (Marsland, 2014). Consequently, repeated initialisations are required.<br><br>3. The number of clusters needs to be determined before the algorithm can be implemented. Consequently, repeated initialisations are required.<br><br>4. Overfitting is possible with too many clusters.<br><br>5. It is sensitive to noisy data or outliers. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **KMC** (cont.) | | 6. An assumption is made that the clusters are isotopic and convex, which is not always true (Pedregosa *et al.*, 2011).<br><br>7. The curse of dimensionality is evident with Euclidean distances. The curse of dimensionality describes the growing problem which arises when more and more features are added to a dataset. The features are added to increase the generalisation capability of the model; however, it comes at the cost of an increasingly slow learning model (Nisbet *et al.*, 2009). |
| **MS** | 1. It does not make model assumptions and therefore it can model non-convex shaped clusters.<br><br>2. It has only one input parameter.<br><br>3. It does not have a local minimum and the region size determines unique clustering without requiring repeated initialisations.<br><br>4. It is relatively insensitive to outliers.<br><br>5. The trained model can be stored. | 1. There is no control over the number of clusters determined (Carreira-Perpiñán, 2015).<br><br>2. It is computationally expensive. |

#### 4.4.1.7 The different clustering performance metrics

According to Pedregosa *et al.* (2011), there are various metrics to evaluate the performance of a clustering algorithm, including the adjusted rand index, mutual information

based scores, homogeneity, completeness, V-measure, Fowlkes-Mallows scores, silhouette coefficient, Calinski-Harabaz index, Davies-Bouldin index and contingency matrix. The mathematical formulations are available in Pedregosa *et al.* (2011).

Some metrics require ground truth class assignments (GTCA), which are the true labels or cluster assignments of the independent variable(s). Ground truth classes are hardly available in practice. It can be manually assigned by human annotators or subject matter experts (SMEs) of the industry or area that the dataset originates from (Pedregosa *et al.*, 2011).

Metrics which require GTCA will be presented first. For all of them, excluding the contingency matrix, the best possible value which indicates perfect labelling, good similarity and significant agreement between two clusters is the value of 1. Metrics which require GTCA are as follows:

1. Adjusted rand index

   The adjusted rand index (ARI) determines the similarity between the predicted output of the model and the GTCA. It ignores permutations and it performs chance normalisation. It determines the number of pairs of instances which are in the same set in both the GTCA and the predictions, and the number of pairs of instances which are in different sets in both the GTCA and the predictions. It divides this by the total number of possible pairs in the dataset to normalise the metric. The metric is adjusted to improve its performance. A value close to 0 indicates random uniform label assignments.

2. Mutual information-based scores

   The mutual information-based (MI) score determines the agreement between the predicted output of the model and the GTCA. It ignores permutations as well. It uses the probabilities of an instance occurring in a cluster, class and in the intersection of the cluster and class. Three versions are possible: as-is MI, normalised MI (NMI) and adjusted or normalised against chance MI (AMI). A value close to 0 indicates random, independent, uniform label assignments (not true in the case of the as-is score).

3. Homogeneity

   Homogeneity ($h$) measures whether a cluster consists of instances which only belong to a single class.

4. Completeness

   Completeness ($c$) measures whether all members of a specified class belong to the same cluster.

5. V-measure

   The V-measure is the harmonic mean of both homogeneity ($h$) and completeness and the formula is as follows: ($c$), $V = \frac{2hc}{h+c}$.

6. Fowlkes-Mallows score

   The Fowlkes-Mallows index (FMI) or score, measures the geometric average of the pairwise precision and recall (explained at the classification metrics). A value close to 0 indicates random, independent, uniform label assignments.

7. Contingency matrix

   The contingency matrix determines the set of instances at the intersection for every true ground truth/predicted pair, by plotting the true clusters to the predicted clusters. The contingency matrix is based on the assumption that instances are independent and identically distributed. It is similar to the confusion matrix used in classification.

Clustering metrics which do not require GTCA are as follows:

1. Silhouette coefficient

   The silhouette coefficient measures how similar an instance is to its own cluster compared to other clusters, based on distance measurements. The higher the value, the denser the clustering model with better-defined, dense and well-separated clusters. It is determined by using two distance measures: mean intra-cluster and mean nearest-cluster distances. The intra-cluster distance of an instance is subtracted from its nearest cluster distance and divided by the maximum distance of the two measurements to normalise the value. The mean silhouette coefficient is given for a dataset (a set of instances). The value ranges on $(-1, 1)$ where higher values are preferred since they indicate similarity. Values near 1 indicate good clustering and *vice versa* for values near -1. Values near 0 indicate overlapping clusters and negative values indicate that an instance is clustered in the wrong cluster since a different cluster is more similar to it.

2. Calinski-Harabaz index or variance ratio criterion
   The Calinski-Harabaz value is determined by using the ratio of inter-cluster dispersion and intra-cluster dispersion. The goal is to achieve a high Calinski-Harabaz score, which indicates dense and well-separated clusters.

3. Davies-Bouldin index
   The Davies-Bouldin index calculates the average similarity between each cluster and its most similar cluster, by using the distance between the centroids of the clusters as well as the distances between the instances of a cluster and its centroid. The lowest and best possible score is 0 which indicates the best separation between clusters.

For the purposes of this research study, FMI has been chosen as the clustering performance metric, since it is a commonly used metric, is simplistic in nature and easy to understand.

### 4.4.2 Classification algorithms

The following classification algorithms will be covered by the study: DTs, LogReg, KNNs, NB, NNs, RFs and SVC. These algorithms are most popularly used in practice and are well supported in *Python*. Table 4.5 presents the classification algorithms with the following characteristics for each: 1) the source where they can be researched in detail, including their mathematical formulations, 2) applications found in literature for a variety of data types, including textual data, image data, video data, audio data, time-series data and transactional data, and 3) the application purposes as mentioned in Subsection 3.1.2.

#### 4.4.2.1 Decision trees

A DT is a hierarchical data structure which identifies local regions by implementing the divide-and-conquer strategy using a series of recursive splits. It consists of the base (root), internal decision nodes (forks), paths between the decision nodes (branches) and terminal leaf nodes (leaves).

Table 4.5: A summary of the applications of classification algorithms

| Algorithm | Source | Data Types | Applications |
|---|---|---|---|
| All | | | Classification |
| All DTs | | | Educational assistance<br>Reverse prediction<br>Simulation<br>Summation<br>Visualisation - tree diagram |
| CART | Alpaydin (2010)<br>Gupta *et al.* (2017)<br>Harrington (2012)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Textual data (Jotheeswaran & Kumaraswamy, 2013)<br>Image data (Lawrence & Wrlght, 2001)<br>Video data (Patsadu *et al.*, 2012)<br>Audio data (Wu *et al.*, 2010)<br>Time-series data (Pesch *et al.*, 2011)<br>Transactional data (Sathyadevi, 2011) | Time-series prediction |
| ID3 | Alpaydin (2010)<br>Gupta *et al.* (2017)<br>Harrington (2012)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Textual data (Harrag *et al.*, 2009)<br>Image data (Oliver *et al.*, 2005)<br>Video data (Glasberg *et al.*, 2008)<br>Audio data (Glasberg *et al.*, 2008)<br>Transactional data (Sathyadevi, 2011) | |
| RF | Marsland (2014)<br>Pedregosa *et al.* (2011) | Textual data (Ali *et al.*, 2012)<br>Image data (Khan *et al.*, 2010)<br>Audio data (Phan *et al.*, 2015)<br>Time-series data (Tatsumi *et al.*, 2015)<br>Transactional data (Coussement & den Poel, 2009) | Outlier detection<br>Time-series prediction |
| All variants of KNNs | | | Educational assistance<br>Outlier detection<br>Simulation |

Continued on next page

| Algorithm | Source | Data Types | Applications |
|---|---|---|---|
| | | | Visualisation |
| KNN | Harrington (2012)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Textual data (Lin *et al.*, 2014; Tan, 2005)<br>Image data (Ramteke & Khachane, 2012)<br>Video data (Babu & Ramakrishnan, 2004)<br>Audio data (El-Maleh *et al.*, 2000)<br>Time-series data (Chaovalitwongse *et al.*, 2007)<br>Transactional data (Choi *et al.*, 2010) | Time-series Prediction |
| NC | Pedregosa *et al.* (2011) | Textual data (Tan, 2005)<br>Image data (Rosdi *et al.*, 2015)<br>Audio data (Tamatjita & Mahastama, 2016)<br>Transactional data (Pedregosa *et al.*, 2011) | |
| RNN | Pedregosa *et al.* (2011) | | |
| LogReg | Harrington (2012)<br>Ng & Stanford University (2016)<br>Pedregosa *et al.* (2011) | Textual data (Zhang & J. Oles, 2001)<br>Image data (Cheng *et al.*, 2006)<br>Video data (Batra *et al.*, 2008; Wu *et al.*, 2018)<br>Audio data (Laurier *et al.*, 2008)<br>Time-series data (Buciumas & Priestly, 2016)<br>Transactional data (Kahanda & Neville, 2009) | Educational assistance<br>Simulation<br>Summation<br>Time-series prediction<br>Visualisation |
| MLP | Alpaydin (2010)<br>Marsland (2014) | Textual data (Tas & Gorur, 2007)<br>Image data(Khan *et al.*, 2010)<br>Video data (Babu & Ramakrishnan, 2004)<br>Audio data (Iliou & Anagnostopoulos, 2010)<br>Time-series data (Nanopoulos *et al.*, 2001)<br>Transactional data (Paliwal & A. Kumar, 2009) | Simulation<br>Time-series Prediction |
| All NBs | | | Simulation<br>Visualisation |

Continued on next page

| Algorithm | Source | Data Types | Applications |
|---|---|---|---|
| BNB | (Pedregosa *et al.*, 2011) | Ttextual data (McCallum & Nigam, 1998) | |
| CNB | (Pedregosa *et al.*, 2011) | Textual data (Rennie *et al.*, 2003) | |
| GNB | Marsland (2014)<br>(Pedregosa *et al.*, 2011) | Textual data (LI & Jain, 1998)<br>Image data (Deselaers *et al.*, 2005)<br>Video data (Sebe *et al.*, 2002)<br>Audio data (Eghbal-zadeh *et al.*, 2015)<br>Time-series (inapplicable due to assumptions)<br>Transactional (Pedregosa *et al.*, 2011) | Time-series Prediction |
| MNB | (Pedregosa *et al.*, 2011) | Textual data (McCallum & Nigam, 1998) | |
| All SVCs | | | Data compression<br>Educational assistance<br>Outlier detection<br>Simulation<br>Time-series prediction<br>Visualisation |
| Linear SVC | (Alpaydin, 2010)<br>(Pedregosa *et al.*, 2011)<br>Cortes & Vapnik (1995)<br>MIT OpenCourseWare (2014)<br>Chang & Lin (2011)<br>Smola & Schölkopf (2004) | Textual data (Kim *et al.*, 2005; Zhang & J. Oles, 2001)<br>Image data (Kim *et al.*, 2002; Lyu & Farid, 2003)<br>Video data (Babu & Ramakrishnan, 2004)<br>Audio data (Garcia-Romero & Espy-Wilson, 2010)<br>Time-series data (Eads *et al.*, 2005)<br>Transactional data (Niimi, 2015) | |
| SVC with kernel | | Textual data (Kim *et al.*, 2005)<br>Image data(Khan *et al.*, 2010)<br>Video data (Babu & Ramakrishnan, 2004; Patsadu *et al.*, 2012)<br>Audio data (Laurier *et al.*, 2008; Lu *et al.*, 2003)<br>Time-series data (Kampouraki *et al.*, 2009)<br>Transactional data (Niimi, 2015) | |

Each decision node implements a test function to its input which outputs discrete outcomes corresponding to the labels of the branches. The input is split based on these discrete outcomes (branches) and each subset moves to the next decision node. The process begins at the root and is replicated recursively through the decision nodes until a leaf node is reached. The final output is the value written on the leaf, which can be a class code or a numeric value. Thus, DTs can be used for both classification and regression applications.

Each test function is a simple function which defines a discriminant in the $d$-dimensional input space which divides the input space into smaller regions. When the path is taken from the root through the decision nodes, the regions become repeatedly subdivided until a leaf node is reached. A leaf node represents a confined region (boundaries are determined by the test functions which lead to this leaf) in the input space where all instances belonging to this region have the same class codes or labels (classification) or nearly identical numeric outputs (regression). These repeated test functions enable the simplification of a complex function or input to a series of simple decisions.

The goal is to find the smallest DT; however, it is an NP-complete problem. Thus, local search heuristics are used to determine the DTs (Alpaydin, 2010). In general, DTs are trained with greedy learning algorithms where at each decision node all possible splitting options are evaluated and the best split at that point is chosen regardless of its effects later on in the development process and tree structure (Marsland, 2014).

A DT is an efficient non-parametric, supervised learning method. In parametric methods a model is defined over the whole input space and its parameters are determined or learnt from the training data. Thereafter the model and its parameters are applied to the test input. In non-parametric methods a model is defined per local region, where the local regions are determined by dividing the input space according to a distance measure, for example, the Euclidean norm. Each local model is trained on the training data from the region where it belongs. When presented with test data the local region is determined and the corresponding model is applied to the data. The DT is a non-parametric method: it grows during learning (branches and leaves are added) since its structure is not determined before the process starts. Another reason is that no parametric form is assumed for the class densities of the input data.

The structure of the DT can be visualised with a decision tree diagram which indicates the root, decision nodes and their conditions, the branches and the leaves.

For classification problems the goal of the DT is to create pure leaves where only one class reaches a leaf node. However, when the labels of the data reaching the leaf node are impure (mixture of classes), a majority vote is made on the class to which these data points belong. The output of this vote is the value of the leaf node. In regression applications the value of the leaf node is the average output value of the data reaching the leaf node.

Important concepts regarding DTs include tree induction, tree size, DT complexity, stopping criterion, feature selection process types, inductive bias, branching factor, information entropy, information gain, the Gini index, complexity parameter, pruning, pre-pruning and post-pruning. Details are available in Harrington (2012); Marsland (2014) and Alpaydin (2010).

Different types of DTs are available, including binary trees, univariate trees, multivariate trees, classification trees, regression trees and CARTs (Alpaydin, 2010; Marsland, 2014). The trees relevant to this research study are as follows:

1. Classification trees

   Classification trees are used for classification applications where leaf nodes have discrete values or labels. The *impurity measure* measures the quality or goodness of a split at a decision node. A split is pure if after the split all the instances belong to the same class and a leaf node labelled with the class to which all the instances belong can be added (Alpaydin, 2010). If a split is not pure then its instances should be split based on a feature which results in minimised impurity after the split since it will lead to the development of the smallest tree. There is much emphasis on the impurity measure since it aids in developing the smallest DT, which is the goal of the DTs. There are various methods available to measure the impurity, including information entropy, information gain and the Gini Index. Examples of classification trees are the iterative dichotomiser 3 (ID3) and C4.5 (an extension of the ID3).

2. Regression trees

   Regression trees are used for regression applications where leaf nodes have numeric values. Regression trees are the same as classification trees; however, the only

difference is the impurity measure which needs to be adjusted for numeric values. The impurity measure is determined by the mean squared error (MSE), mean absolute error (MAE) or the worst possible error. The class labels are replaced with averages and the error is calculated as the sum of the differences between the output values and the mean of the output values of the instances at the decision node (Pedregosa *et al.*, 2011).

A leaf node is created when the error is below a pre-specified threshold value. If the error is above the threshold value, another decision node is created to split the instances further. At each decision node the feature and the split threshold (only for numeric features) resulting in a split with the lowest error and largest reduction in error from before to after the split, is chosen.

Instead of using the averages as the leaf values, linear regression fitted over the instances at the leaf can be used. The linear regression fit creates smaller trees at additional computation cost (Alpaydin, 2010).

3. Classification and Regression trees

   The CART algorithm is a tree-based algorithm for both classification and regression applications. Classification is performed by creating binary trees and using the Gini index or Gini impurity as the impurity measure. Binary trees split only one feature at a time. Regression is performed by using the sum of squares error (SSE) as the impurity measure. The CART algorithm has a preprocessing step to decrease the node complexity and reduce dimensionality through subset selection. A multivariate version of CARTs is also available (Alpaydin, 2010). A disadvantage of CARTs is that it is possible to repeatedly use features at splits and therefore the number of features does not decrease after each split (Harrington, 2012).

#### 4.4.2.2 $k$-nearest neighbour

The KNN algorithm is a supervised learning method used to classify data (Harrington, 2012). It arranges the labelled input data and the new data point in the input vector space according to their coordinates (the data point values) and calculates the distance from each input point to the new point. The $k$ nearest data points are the neighbours of the new data point and they are used to determine the class of the new point

(Marsland, 2014). For classification a majority vote from the classes of the neighbours is used to determine the label of a new point. The KNN algorithm is simple and effective; however, it is computationally expensive since the entire input dataset is used for the classification of every new data point. Therefore, KNN is not trained and does not learn from the data. It provides instance-based learning since instances of the data are needed to perform the algorithm.

The variable $k$ is an integer value specified by the user and is usually less than 20. A small value for $k$ increases the flexibility of the algorithm at the cost of higher variance since little data is available on which to base the classification. A small value for $k$ creates a model which is sensitive to noise (Harrington, 2012). A large value for $k$ decreases the accuracy since data points far away from the new point are considered when classifying the new data point. With a large value for $k$ the variance decreases, the model is less flexible and it has more bias (Marsland, 2014).

Traditionally, the nearest neighbour distances are calculated using the brute force method where the distances between all pairs of points in the input dataset are calculated. It is an inappropriate method for large input datasets or datasets with high dimensionality (many features). In order to address the computational expense of the KNN algorithm, methods have been suggested to enable efficient distance calculations, namely using DTs. These methods include the KD-tree and the ball tree.

The KD-tree ($k$-dimensional tree) constructs a tree of binary trees which repeatedly splits one feature or dimension at a time and constructs a line through the median of the coordinates of the split dimension (Marsland, 2014). The KD-tree is very fast; however, it may become inefficient with large dimensionality (Pedregosa *et al.*, 2011).

The ball tree initially constructs two balls or spheres (multidimensional), where each instance belongs to only one ball, based on the shortest distance to the centroid of the ball. The balls may overlap. Binary trees are constructed to repeatedly split each ball into two balls and the data points are allocated to one of the new balls based on the shortest distance to the centroid of the new ball. The ball-tree outperforms KD-trees with high dimensionality datasets, depending on the structure of the input dataset (Pedregosa *et al.*, 2011).

A variety of distance measures are available, including Euclidean, Minkowski and Manhattan (city block) distance (Marsland, 2014).

Variants of KNN include the radius nearest neighbour classifier (RNN), nearest centroid classifier (NC), $k$-nearest regressor or $k$-neighbours regressor (KNR) and radius neighbour regressor (RNR).

The RNN uses the radius (specified by the user) to determine the neighbours of the new data point which are then used to classify the new data point based on majority vote. The NC presents every class by the centre or centroid of its members. A new data point is assigned to the class where the distance from it to the centroid is the shortest. An advantage of this algorithm is that is has no parameters which need to be specified or determined by the user. A disadvantage is that it makes an assumption that the variance is equal in all dimensions, which leads to poor performance on non-convex classes (Pedregosa *et al.*, 2011). The NC is only applicable for classification.

The KNN and RNN can be adjusted to accommodate regression by using the average of the labels of the neighbours instead of the majority vote of the classes to determine the prediction of a new data point. This leads to the KNR and RNR algorithms.

#### 4.4.2.3 Logistic regression

The LogReg algorithm is a supervised learning technique which is used for classification using numeric or nominal values (Harrington, 2012). It is designed for binary classification and can be adjusted for multi-class classification problems. It is also called the 'logit regression', 'maximum-entropy model' or the 'log-linear classifier' (Pedregosa *et al.*, 2011).

The goal is to determine the optimally separating decision boundary which is defined by the parameters or the weights of the LogReg. The algorithm starts by multiplying the input dataset by weights, which include a bias to account for random noise. The result is a decision boundary which distinguishes between two classes (a positive and a negative class). Next, the result is fed to the sigmoid function, namely a non-linear function, which outputs the probability of an instance belonging to one of the two classes. This is called the objective function of LogReg. A probability greater or equal to 0.5 indicates that the used instance belongs to the positive class ($y = 1$), else it indicates that it belongs to the negative class ($y = 0$).

Optimisation algorithms (also called 'solvers') are used to determine the optimal weights to ensure the best classification takes place on the current dataset (Harrington,

2012). Gradient descent is most commonly used. Other solvers include conjugate gradient descent, stochastic gradient descent (SGD), Broyden–Fletcher–Goldfarb–Shanno (BFGS) and limited memory BFGS (LBFGS) (Ng & Stanford University, 2016). *Python* implements these solvers together with stochastic average gradient descent (SAG), SAGA (variant of SAG), truncated Newton methods (Newton-CG or NTCG methods) and LIBLINEAR (a library for large-scale linear classification) (Pedregosa *et al.*, 2011). The mathematical formulations of the solvers and how the solvers determine the best-fitting weights are available in Ng & Stanford University (2016).

Due to the nature of the optimisation problem, the weights have to be determined iteratively. The chosen solver is repeatedly applied until a stopping criterion is met. This means that a predetermined number of iterations have been reached, the classification success is above a specified threshold or the classification is within a specified tolerance margin (Harrington, 2012).

To enable multi-class classification, one-vs-rest (OVR), also called one-vs-all (OVA), classification is mostly used. If there are $k$ different classes, $k$ different binary classifications will be performed. Each class gets a turn to be the positive class and the rest of the classes are grouped together to create the negative class (Pedregosa *et al.*, 2011). Each of the $k$ classifiers learns to identify its own positive class. When a new data point is classified, all $k$ classifiers are used and the classifier with maximum probability indicates the class to which the new data point belongs.

#### 4.4.2.4 Naïve Bayes

The NB algorithm is a supervised learning method which is based on Bayes's theorem. It is a subset of Bayesian decision theory and performs classification tasks. Bayes's theorem includes the following probabilities: $P(C_1|X)$ is the conditional probability (the probability of class 1 given the input data), the prior probability is $P(C_i)$, the posterior probability is $P(C_i|X_j)$, and the class-conditional probability is $P(X_j|C_i)$ (Marsland, 2014).

The prior and class conditional probabilities are determined from the training set and used to determine the posterior probability. A new data point is assigned to the class according to the maximum a posteriori (MAP), which is the probability which maximises the posterior probability. MAP indicates the most likely class of new instances given the history or training data.

An important assumption is made regarding the data, to elevate the effect of increasing dimensionality, namely that the features are conditionally independent, given the classification (Marsland, 2014).

There are a variety of different Naïve Bayes classifiers available, which differ in the assumptions made regarding the distribution of $P(x_i|y)$. The Gaussian NB (GNB) assumes the likelihood of features to be Gaussian. The multinomial NB (MNB) is designed for multinomially distributed data and is typically used for text classification. The complement NB (CNB) is an alteration of the MNB to accommodate imbalanced input datasets sets. It is also suitable for text classification. The Bernoulli NB (BNB) assumes that the input data is distributed according to multivariate Bernoulli distributions. Input data and feature vectors should be binary-valued for this algorithm. It is also suitable for text classification using word occurrence vectors instead of word count vectors (Pedregosa *et al.*, 2011).

#### 4.4.2.5 Neural networks

A neural network (NN) or artificial neural network (ANN) is an arrangement of statistical algorithms whose structure is based on the biological brain patterns found in human brains. NNs are used to identify and create the non-linear relationships between input variables and the output variable(s).

An NN consists of an input layer, hidden layer(s) and an output layer. The input layer provides the input variables. The hidden and output layers consist of perceptrons or neurons which perform weighted and biased summations of their inputs, which are passed to mathematical functions called activation functions. The results of the activation functions are then sent to the next layer. The neurons in the hidden layers are interconnected, in parallel and their weights are determined experimentally (iteratively over time). The number of neurons in the output layer is equal to the number of objective functions or dependent variables. Figure 4.4 illustrates the basic NN.

Typical NN parameters include the number of hidden layers, number of neurons per layer (from the first hidden layer to the output layer), number of delays or delay units in a recurrent NN (number of hidden layer outputs which are fed back to the input of the hidden layer), learning rate, the activation function and the momentum rate (Alpaydin, 2010; Marsland, 2014).

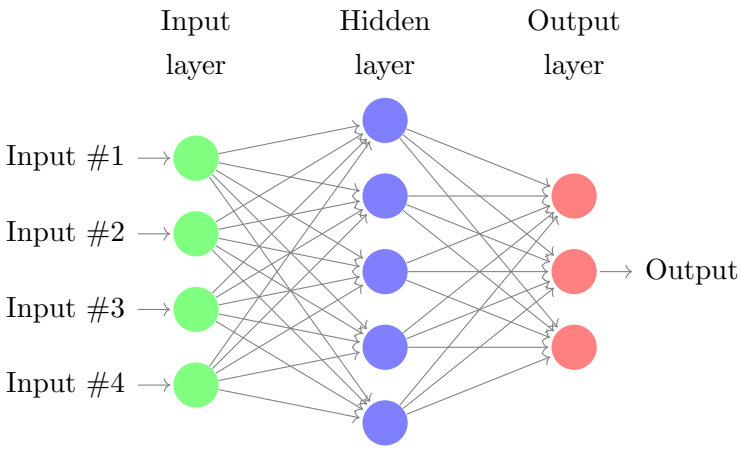


Figure 4.4: The basic neural network (Neural netwotk image, 2018)

Various trade-offs exist with NNs; usually they involve the quality-time-cost triangle. For example, when increasing the number of neurons, the precision increases (quality); however, the model development time increases as well. The following are the most prominent parameters:

1. The number of hidden neurons

   Too few or too many neurons lead to poor generalisation, where too few neurons cannot be effectively trained and too many neurons memorise the noise in the training set. However, if the number of hidden neurons increases, it becomes more difficult to train the model (Alpaydin, 2010). The optimal number of hidden neurons can be determined experimentally or via cross-validation.

2. The number of hidden layers

   Theoretically the number of hidden layers is endless; however, it comes at a computational cost. It is proven mathematically by the Universal Approximation Theorem that one hidden layer with various neurons is sufficient for any problem (Marsland, 2014).

3. The learning rate

   The learning rate, $\eta$ (also called the 'convergence rate'), determines the adjustment size of the weights in the training phase, meaning how quickly the network learns (Attoh-Okine, 1999). If it is too small, the NN takes a long time to reach a local optimum (converge). If it is too large, the NN overshoots or misses the local

optima, oscillates and can fail to converge. It may even diverge. The learning rate also determines the dependency on recent instances. If it is too large, the dependency is large and *vice versa* (Alpaydin, 2010).

4. The momentum rate
   The momentum rate, $\alpha$, influences the training speed of the NN. It determines the relative contribution of the current and past weight changes to the current weight change, by adding a proportion of the previous weight change to the current weight changes. Thus, it influences the size of the weight adjustment (Attoh-Okine, 1999).

The goal of the model is to determine and optimise the weights and biases associated with each neuron to minimise the error function between actual and desired output (Corne *et al.*, 2016). The training process continues until the error stops decreasing or a pre-specified range is reached (Boschetto *et al.*, 2013). The NN has to be trained multiple times due to the dependence on the initial weight values, which influences the error.

Optimisation algorithms are used to determine the optimal weights and biases to ensure the best classification takes place on the current dataset ((Harrington, 2012). A training or optimisation algorithm determines the change in the weight of a neuron (Sadeghkhani *et al.*, 2012). A variety of training algorithms for NNs are available, including back-propagation (BP), back-propagation through time (BPTT), gradient descent algorithms, Newton's method (Newton-Raphson method), Quasi Newton method, Levenberg–Marquardt training algorithm, Hessian, BFGS, L-BFGS, Hessian diagonal, momentum, Delta-bar-delta technique, SuperSAB algorithm, clip the logistic activation function, Adadelta, adaptive moment estimation (ADAM), AdaMax, AdaGrad, resilient backpropagation (RPROP) (Riedmiller & Braun, 1993), iRPROP, RMSProp and the Robbie Jacobs adaptive weights method. Gradient descent is further divided into traditional or full batch gradient descent (BGD), stochastic gradient descent (SGD), minibatch gradient descent, normal gradient descent (NGD) (Amari, 1998), conjugate gradient descent (CGD) and vSGD (Alpaydin, 2010; Bottou, 2012; Hinton *et al.*, 2012; Kingma & Ba, 2015; Marsland, 2014; Pedregosa *et al.*, 2011). Gradient-based training algorithms are popularly used.

Activation functions play an important role in the generalisation capability of NNs. Activation functions are also called processing, transfer, basis or neuron function(s). Activation functions can be categorised according to the output they provide as well as their shape. Different activation functions are used for classification and regression problems. Activation functions include the identity or linear function, binary step function, bipolar step function, logistic or sigmoid function (further divided into the binary sigmoidal function and bipolar sigmoidal function), ramp function or rectified linear unit (ReLU), leaky ReLU, periodic functions (sine, cosine, tangent, cosecant, secant and cotangent functions), radial basis function, exponential function and hyperbolic functions (hyperbolic sine, cosine, cosecant, secant and cotangent).

Important concepts for NNs are deep learning, online learning, offline learning, batch training, sequential training, temporal association tasks, NN complexity, Bayesian Inference (BI), early stopping generalisation, adaptive learning and Bayesian regularisation (Alpaydin, 2010; Conde *et al.*, 2018; Marsland, 2014).

Neural networks can be categorised in various ways, depending on the type of neurons used, the connections between neurons in the network or the overall structure of the network. The two main types of NNs are feed-forward NNs (FF-NNs) and recurrent NNs (R-NNs). In FF-NNs the data flows unidirectionally from the input layer, through the hidden layers(s) to the output layer. FF-NNs include the single layer perceptron (SLP), the basic FF-NN, the multilayer perceptron (MLP), radial basis function neural network (RBFNN), autoencoder (AE), variational autoencoder (VAE) and the sparse autoencoder (SAE) (Alpaydin, 2010; Bengio, 2009; Boschetto *et al.*, 2013). Recurrent NNs (R-NNs) employ at least one feedback loop. The R-NN is more suited for dynamic evolutions (Conde *et al.*, 2018). R-NNs include the basic R-NN, Elman based R-NN (ERNN), Jordan based R-NN (JRNN), long short-term memory network (LSTM) and the gated recurrent unit (GRU) (Carcano *et al.*, 2008; Conde *et al.*, 2018; Şeker *et al.*, 2003; Tai *et al.*, 2015). Other NNs include the modular NN (MNN), extreme learning machine (ELM), liquid state machine (LSM), echo state network (ESN) and the support vector machine (SVM).

#### 4.4.2.6 Random forests

The RF algorithm is a type of ensemble method where multiple identical learners with variances or randomness are combined to produce better results than a single learner or

model. Variances are included to make the learners slightly different from one another. The basic idea of RFs is that if a single tree is good, then a forest of trees (many trees) should be better, given that there is variety between them.

The RF algorithm creates randomness from a standard dataset by utilising different methods, including bagging and limited feature selection. Bagging is the process of taking bootstrap samples from the dataset to create slightly different training datasets for every tree. Bootstrap samples are samples which are the same size as the whole dataset but are taken from the whole dataset with replacement. Different trees learn on the different datasets and together they create the random forest. To increase the introduced randomness, limited feature selection is used, where every decision node is limited to a random subset of the features of the dataset. This limits the choices that the decision tree can make and enables the decision trees to learn faster.

These two methods of introducing randomness lead to reduced variance without affecting the bias (Marsland, 2014). The bias of the forest increases slightly (with reference to the bias of one non-random tree). Due to averaging, its variance also decreases which compensates more for the slight increase in bias which results in an overall better model (Pedregosa *et al.*, 2011). The averaging of the outputs of the DTs also improves the prediction accuracy and the control over the overfitting of the RF.

Two parameters are required for RFs, namely the limited feature selection size and the number of trees in the forest. The size of the subset of features to consider at each decision node is usually set to the square root of the total number of features which are available for classification applications (Marsland, 2014). For regression applications it is set to the total number of features which are available. The smaller the size of the subset, the greater the decrease in variance and the greater the increase in bias (Pedregosa *et al.*, 2011).

An increase in the number of trees leads to an increase in RF accuracy at the cost of more computation time (Pedregosa *et al.*, 2011).

The algorithm methodology works as follows. For each of the trees the following is performed (Marsland, 2014):

1. Choose a new bootstrap sample from the training dataset.

2. Train a DT using this bootstrap sample.

3. At each decision node of the tree, randomly determine the subset of $m$ features and use it to determine the impurity measure at the node. Select the optimal feature to continue from, based on the impurity measure.

4. Repeat steps 1 to 3 until a full tree is developed.

The process of building trees continues until the error stops decreasing (Marsland, 2014). The impurity measure or criterion is the same as for DTs. For classification, the entropy, Gini or misclassification measures could be used and for regression the mean squared error (MSE) or mean absolute error (MAE) could be used (Pedregosa *et al.*, 2011).

After the trees have been built (trained), the majority vote for classification applications and the average response for regression applications is the output of the forest (Marsland, 2014).

*Python* also introduces another type of tree and forest, called an extra tree (ET) and extra forest (EF), respectively. The ET fits a number of randomised DTs on different random subsets of the input dataset and uses averaging to increase the accuracy and control overfitting. The extra forest creates highly randomised trees. There is added randomness with the splitting rule compared to that of the RF. In general, RFs search for the most discriminative threshold in the random subset of candidate features, where the EF randomly draws thresholds for each candidate feature and the best is used as the splitting rule. The EF has decreased variance at the cost of a slightly increased bias (Pedregosa *et al.*, 2011).

#### 4.4.2.7 Support vector machines

The SVM or support vector network was invented for two group classification problems where the training data is linearly separable and labelled, making it a supervised learning technique (Cortes & Vapnik, 1995). It is also called a 'discriminative classifier' since it discriminates between classes. Throughout the years different versions were developed which can accommodate non-linear separable training data, regression problems, clustering problems, anomaly detection (unsupervised learning), multi-class classification and semi-supervised learning cases.

The goal of the SVM is to determine the best hyperplane which ensures the optimal separation of the two classes by maximising the margin of the training data. By

maximising the margin, the data is correctly classified and the best generalisation is achieved (Alpaydin, 2010). The margin is the distance from the data closest to the hyperplane on its left-hand side to the data closest to the hyperplane on its right-hand side. Thus, it is approximately twice the distance to the closest point to the hyperplane. The hyperplane is defined as a linear decision function which separates data in two groups since it is linear. The dimension of the hyperplane is always one less than the dimension of the data: for example, if the data is one-dimensional the hyperplane is a dot; if the data is two-dimensional the hyperplane is a line; if the data is three-dimensional the hyperplane is a plane; if the data is four or more dimensional the hyperplane is a hyperplane.

In order to find the optimal hyperplane, two auxiliary, parallel hyperplanes are constructed, one on each side of the optimal hyperplane and parallel to the optimal hyperplane. Each auxiliary hyperplane goes through the data closest to the optimal hyperplane. Next, the margin is determined by calculating the perpendicular distance between the auxiliary hyperplanes. By maximising the margin, the optimal weights and bias which construct the optimal hyperplane and ensure maximal separation between the two classes are determined.

An optimisation process is employed to maximise the margin. The support vectors are determined through the optimising process. The support vectors are the data points from the input dataset which lie on the margin. They are used to calculate the bias of the optimal hyperplane and to construct it.

A variety of SVMs have been developed, including the traditional SVM or support vector classifier (SVC), soft-margin SVM, linear SVM, SVM with kernel trick, multi-class SVM, Support Vector Regressor (SVR), one-class SVM, Least squares SVM (LS-SVM), ranking SVM (RSVM), structured or structural SVM (S-SVM) and transductive SVM (T-SVM) (Manning *et al.*, 2008; Wang *et al.*, 2006).

The following SVMs are relevant to this study:

1. Traditional SVM

   It is a supervised learning algorithm which uses linearly separable, labelled data to perform two-class classification. It is also called support vector classification (SVC).

2. Soft-margin SVM
   A parameter, the regularisation parameter, is added as a penalty term to the traditional SVM to accommodate a small number of outliers which are in the margin or are misclassified data points.

3. SVM with kernel trick
   This SVM is similar to the traditional SVM; however, it uses non-linearly separable, labelled data and a kernel function or non-linear basis function, which maps input to a usually higher dimensional space where a linear separation is possible and constructed on the transformed data.

4. Linear SVM
   The linear SVM is an SVM which does not use non-linear kernel functions; thus, it seperates the data linearly.

5. Multi-class SVM
   Originally, SVMs were designed for two-class classification and different methods were developed to employ SVMs for multi-class classification.

6. Support Vector Regression (SVR)
   The SVR uses the same method as the SVC to predict real-valued output instead of performing classification. Kernel functions can also be employed.

7. One-class SVM
   This is an SVM for unsupervised learning where high-density areas or regions are estimated. This is usually used for outlier or novelty detection, which is a type of clustering method.

Data is mostly not linearly separable and there is no hyperplane which can separate that data perfectly. Therefore, a hyperplane to accommodate the deviation of the data points from the margin is needed. Deviations take two forms: misclassified data points which lie on the wrong side of the hyperplane and data points lying in the margin on the correct side of the hyperplane (correctly classified) (Alpaydin, 2010). The data points satisfying the deviations are called the non-separable points. The soft-margin SVM accommodates the deviations by using one of two possible implementations: using a regularisation parameter $C$ or regularisation parameter $v \in (0, 1]$.

The SVM with kernel trick is applied when the data has too many outliers and the classification error is too large. The idea is to map the data to a new space, usually a higher dimensional space than the current one, by performing non-linear transformation using a chosen basis function and then creating a linear model in the new space. A hard- or soft-margin SVM could be used as the base. Three kernel or basis functions are commonly used (Pedregosa *et al.*, 2011): polynomial function, radial basis function (RBF) or hyperbolic tangent function (also called the tanh or sigmoid function). Other variations of kernel functions are also possible. A precomputed kernel could be specified by the data analyst, which is an application specific kernel. Multiple kernel learning is when kernel functions consisting of combinations of different kernel functions are used.

The SVM algorithm is designed for binary classification and generally three methods are employed for multi-class SVM classification.

The methods for multi-class SVM classification are:

- One-versus-all (OVA) classifiers
  If there are $n$ classes, $n$ SVCs are created. One class is chosen as the positive class and the rest of the classes form the negative class. The class of a new data point is determined by the classifier which outputs the highest absolute value (Manning *et al.*, 2008).

- One-versus-one (OVO) classifiers
  This enables the pairwise separation of classes, by creating $\frac{N(N-1)}{2}$ SVCs where two classes are used per SVC. The training time may decrease since the training dataset per classifier is smaller than the one-versus-all method. The class which is chosen by most classifiers is the final class of a new data point (Alpaydin, 2010).

- The Crammer and Singer method
  This method provides a single optimisation problem for multi-class problems by adjusting the constraint to the objective function (Pedregosa *et al.*, 2011).

SVC predicts categorical labels, while SVR predicts real-valued labels. A hyperplane is also constructed for SVR; however, it outputs numerical values. Instead of optimally separating the classes, the hyperplane has to fit the data perfectly. The goal is to determine the weights and bias such that the hyperplane best fits the data with minimised error (Alpaydin, 2010). The optimisation process is the same for SVR as it is

for SVC. The kernel trick can also be incorporated into SVR for cases with non-linearly separable data. An equivalent $C$-SVC and $v$-SVC formulation is available for SVR.

The one-class SVM or support vector clustering is most commonly used for anomaly detection which includes outlier and novelty detection. The one-class SVM is more specifically designed for novelty detection but can be implemented for outlier detection. In outlier detection (unsupervised anomaly detection) the training data contains outliers (data points which are relatively far from the others in the same set) and the model tries to determine areas where the training data is mostly concentrated. In novelty detection (semi-supervised anomaly detection) the training data does not contain any outliers and the model tries to determine whether a new data point is an outlier or not. If it is an outlier, it is called a novelty (Pedregosa *et al.*, 2011). The one-class SVM is specifically designed for detection; however, the other algorithms identify outliers as well.

The one-class SVM constructs a sphere of radius $R$ and with a centre $a$ with two objectives: 1) enclose as much as possible density or instances by finding; 2) the smallest possible radius. It can also be adjusted to accommodate the deviation of the data points which lie outside the sphere. A positive result indicates that the data point lies within the sphere and *vice versa* for a negative result (Schölkopf *et al.*, 2001).

The kernel trick can also be incorporated into the one-class SVM for cases with non-linearly separable data. The kernel functions can be employed by transforming the data to a new space and constructing the sphere there. Thus, arbitrary shapes can be defined in the original space by using a kernel function. An equivalent $C$-SVC formulation is available for the one-class SVM. Another implementation of the multi-class SVM is available, which is widely used by programming packages (Chang & Lin, 2011).

In *Python* six different kernel function options are available: linear (LIN), polynomial (POLY), RBF, sigmoid (SIG), precomputed and callable. Precomputed functions use a specific matrix, the Gram matrix ($G = \mathbf{X} \cdot \mathbf{X}^T$), instead of $\mathbf{X}$ when fitting the model. The callable option enables the user to use a function created in *Python* (Pedregosa *et al.*, 2011).

#### 4.4.2.8 The advantages and disadvantages of the selected classification algorithms

As previously stated, some classification algorithms can be adjusted to accommodate regression applications as well. The underlying algorithm methodology remains the same; however, the output is treated differently. Table 4.6 presents the advantages and disadvantages of the selected purely classification algorithms. Table 4.7 presents the advantages and disadvantages of the selected algorithms which can perform both classification and regression.

Table 4.6: Advantages and disadvantages of the selected purely classification algorithms

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **LogReg** | 1. It is easy to implement (Harrington, 2012).<br><br>2. It is computationally inexpensive.<br><br>3. Gradient descent makes it less complex.<br><br>4. It enables easy interpretation of the knowledge representation (Harrington, 2012). | 1. It easily overfits the data (Harrington, 2012).<br><br>2. It is prone to low accuracy.<br><br>3. The learning rate has to be manually determined or picked.<br><br>4. Gradient descent is a slower optimisation algorithm (Ng & Stanford University, 2016). |
| **NB** | 1. It enables fast computation at low computational cost.<br><br>2. It is simple to implement and can work with small input datasets.<br><br>3. It handles the curse of dimensionality well (Marsland, 2014). | 1. It is a good classifier; however, it is a poor estimator (Marsland, 2014).<br><br>2. It is sensitive to the preparation the input datasets underwent (Harrington, 2012).<br><br>3. An underlying assumption is that the features are conditionally independent. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| | 4. It handles multiple classes well (Harrington, 2012) and enables multi-class prediction applications. | 4. The zero probability problem/frequency problem: if there is no training of a specific class, it leads to a zero posterior probability and the model would be unable to make predictions. |

Table 4.7: Advantages and disadvantages of the selected classification and regression algorithms

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **DTs** | 1. They provide a white box method, namely it is clear what exactly it is doing. Therefore, it is a transparent method. People tend to trust white box applications since they are transparent (Marsland, 2014). | 1. A tree with complex nodes and small size loses the goal of the tree, *i.e* to divide the problem into a set of simple problems (Alpaydin, 2010). |
| | 2. It enables knowledge extraction since it is interpretable and easily understood, which enables human experts to verify the model (Alpaydin, 2010). | 2. With increased noise in the data, larger trees result since there is a strive to grow a pure tree. This may also lead to weak generalisation and overfitting. |
| | 3. They enable feature extraction. It indicates which features, the features at the root, are more important globally. The relevant features are also used by the DT (Alpaydin, 2010). | 3. With increased node complexity, more data is required and overfitting prone trees are produced since less and less data is available as the process moves down the tree. |
| | 4. Trees with simple nodes and low branching factors, are large trees which are more interpretable (Alpaydin, 2010). | 4. Overfitting is possible; thus, overcomplex trees are created which do not generalise the data well. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **DTs** (cont.) | 5. A set of "if-then" rules, or logical disjunctions, can easily be determined from the DT which is suitable for use in a rule induction system (Marsland, 2014). These are also easy to transform into program code. | 5. Many valued dimensions are favoured by splitting, resulting in many branches. Although the impurity is low, the complexity or branching factor is high which is against the principle of splitting to create simple decisions. |
| | 6. There is a low computational cost when developing the tree and an even lower cost to use the tree which enables immediate results and quick responses (Marsland, 2014). | 6. The optimal decision tree cannot be determined since it is an NP-complete problem. The algorithm is based on heuristic algorithms which cannot guarantee the creation of the globally optimal decision tree. |
| | 7. They require little data preparation and they accommodate noisy data and missing data (Marsland, 2014) | 7. If some features or classes dominate, biased trees result. |
| | 8. They indicate the main characteristics of the dataset, including the relevant dimensions and split positions. | 8. Some concepts are difficult to learn since decision trees do not express them easily (Pedregosa *et al.*, 2011). |
| | 9. The learnt model can be validated by using statistical tests, which makes it possible to determine the reliability of the model (Pedregosa *et al.*, 2011). | 9. Small variations in the data may result in unstable decision trees since a completely different tree is being created. |
| | 10. They can be used in both classification and regression applications since they can handle both categorical and numerical data (Pedregosa *et al.*, 2011). | |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **DTs** (cont.) | 11. The derived "if-then" rules are easier to read and understand. Their order in the tree is irrelevant as long as they are accurate in classification (Marsland, 2014).<br><br>12. It is easy to transform the tree into a graph for visualisation purposes (Marsland, 2014).<br><br>13. They can handle multi-output problems (Pedregosa *et al.*, 2011).<br><br>14. The hierarchical structure enables fast localisation of the input space which covers the input. | |
| **CART** | 1. It can perform both classification and regression and can handle both continuous and discrete values, in any combination.<br><br>2. It can determine the interactions among variables (Singh & Giri, 2014).<br><br>3. It can handle missing values and outliers. | 1. It may create unstable decision trees.<br><br>2. It splits by one variable only at a time.<br><br>3. It is non-parametric (Gupta *et al.*, 2017). |
| **ID3** | 1. It enables pruning and the reduction of the number of tests or decision nodes.<br><br>2. Its calculation time is a linear function of the product of the number of features and the number of nodes (Singh & Giri, 2014). | 1. It only performs classification since it is designed for categorical attributes or features, not numeric values.<br><br>2. If it is applied to continuous data it will be computationally expensive and time-consuming, since many trees will be created to determine where to break the continuum (Harrington, 2012). |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **ID3** (cont.) | 3. It produces understandable prediction rules from the data (Gupta *et al.*, 2017).<br><br>4. It searches through the whole dataset to determine the whole tree.<br><br>5. Builds small trees fast. | 3. Overfitting is possible with small datasets.<br><br>4. At decision nodes, one feature is tested at a time which results in a time-consuming process.<br><br>5. In large datasets, ID3 is sensitive to large valued features.<br><br>6. It does not handle missing values well (Gupta *et al.*, 2017). |
| **RF** | 1. They produce highly accurate models and give an estimate of the important variables in classification.<br><br>2. It can handle both continuous and discrete values, in any combination and can perform both classification and regression.<br><br>3. The algorithm is stable. If a new data point is added it may impact one tree; however, it does not influence the overall algorithm.<br><br>4. The trees can be trained in parallel since they are independent from another which reduces the training time.<br><br>5. Cross-validation and pruning are not required (Marsland, 2014)<br><br>6. They deal well with large datasets, missing values or if the data has not been scaled well.<br><br>7. Random forests are not biased. | 1. Sometimes its classification is difficult to be interpreted by humans.<br><br>2. Overfitting is possible with datasets containing noise (Gupta *et al.*, 2017).<br><br>3. They are very complex which leads to an increase in the computational cost and the required training time. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **RF** (cont.) | 8. It recognises outliers and anomalies. | |
| **KNN** | 1. It achieves high accuracy, is robust to noisy data and is effective on large datasets (Harrington, 2012). | 1. It suffers from the curse of dimensionality, namely it is a computationally expensive algorithm since it calculates distances across the full dataset for the classification of each new instance (Marsland, 2014). |
| | 2. It is not sensitive to outliers. | 2. The larger the data, the longer the run-time. |
| | 3. It does not make assumptions about the data and is simplistic and intuitive. | 3. It requires large memory since it stores and uses the full dataset. |
| | 4. It works with both numerical and categorical values. | 4. It does not provide summation of underlying structure of the data. |
| **MLP** | 1. They can approximate the non-linear relationship between independent and dependent variables. | 1. They are like black boxes: they provide results but do not provide explicit information about the underlying processes they incorporate to determine the results (Conde *et al.*, 2018). |
| | 2. They offer high accuracy and consistency of results. | 2. High computational complexity can occur (Corne *et al.*, 2016). |
| | 3. They have good capacity for generalisation (Conde *et al.*, 2018). | 3. They tend to overfit the data, making it more sensitive to noise and approximating the noise in data as well whilst it should be ignored. This increases the number of model parameters. |
| | 4. They can model complex processes. | 4. The empirical nature of developing the model can be viewed as another disadvantage. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **MLP** (cont.) | 5. Any continual and multivariate function can be approximated by an MLP. | 5. It is time-consuming to determine the optimal number of neurons in the hidden layer(s), the learning rate and the momentum rate. The hyper-parameters need to be fine-tuned and are usually determined on a trail-and-error basis (Boschetto *et al.*, 2013). |
| | 6. They enable online learning (Pedregosa *et al.*, 2011). | 6. The MLP has non-convex loss functions due to the hidden layers, which indicates that multiple local minima are possible. Therefore, different validation accuracies are achieved with different random weight initialisations (Pedregosa *et al.*, 2011). |
| **SVM** | 1. They enable sparsity of solution since only the support vectors are needed to solve the problem instead of the entire training set (Alpaydin, 2010). | 1. Overfitting is possible when the number of features is much greater than the number of samples (Pedregosa *et al.*, 2011). |
| | 2. They perform well in cases where the number of attributes or dimensions is greater then the number of instances or samples (Pedregosa *et al.*, 2011). | 2. An expensive five-fold cross-validation method is used to calculate the probability estimates (or ranks for the ranking SVM) since SVMs do not provide it directly. |
| | 3. They have a small number of parameters to specify and can learn non-linear models with the kernel trick. | 3. They can be computationally expensive during the training process (Sapankevych & Sankar, 2009). |
| | 4. The SVM is determined by the sample size, not the dimensionality of the training set. | 4. The specified parameters have to be determined empirically and experimentally. |

Continued on next page

| Algorithm | Advantages | Disadvantages |
|---|---|---|
| **SVM** (cont.) | 5. They enable clustering, classification, regression and ranking. Multi-class classification is possible with the multi-class SVC.<br><br>6. They are effective in high dimensional spaces, are computationally efficient, provide high generalisation and are guaranteed to converge to an optimal solution (Rozenberg *et al.*, 2012; Sapankevych & Sankar, 2009).<br><br>7. Kernel functions enable versatility. | |

#### 4.4.2.9 The different classification performance metrics

For classification output there are three possible outputs as discussed in Section 4.3, and each has specific metrics. Some metrics are versatile and can be applied to more than one type of classification output. Table 4.8 summarises the different performance metrics per classification output. Alpaydin (2010) suggests methods for recomparing different algorithms to each other, including McNemar's Test, k-fold cross-validated paired $t$ test and analysis of variance (ANOVA). The mathematical formulations of the classification metrics are available in Pedregosa *et al.* (2011).

Table 4.8: A summary of the classification performance metrics for the different classification outputs (Pedregosa *et al.*, 2011)

| **Output type** | **Performance metricss** |
|---|---|
| Binary | Precision-recall pairs for different probability thresholds, the receiver operating characteristic (ROC), the balanced accuracy score |
| Binary and Multi-class | Cohen's kappa, the confusion matrix, the average hinge loss, Matthews correlation coefficient |
| Binary, Multi-class and Multi-label | Accuracy, F1 score (or F-score or F-measure), F-beta score, Hamming loss, Jaccard similarity coefficient score, logistic or entropy loss, precision, recall, zero-one classification loss |
| Binary and Multi-label | The average precision, the area under the receiver operating characteristic curve (ROC-AUC) |

The performance metrics for classification include the following:

1. Accuracy

   Accuracy calculates the fraction of samples which have been correctly classified. For multi-label classification, subset accuracies are used and if all the true outputs of the validation dataset are correctly matched by the model for an individual instance, then the subset accuracy is 1, otherwise it is 0. Accuracy is suitable with a balanced dataset.

2. Cohen's kappa

   Cohen's kappa is designed to compare the label allocations of different human annotators instead of a classifier and the GTCA. Cohen's kappa incorporates a probability of expected agreement by chance or the proportion of the times the human annotators would be expected to agree by chance. Cohan's kappa is bounded to the range $(-1, 1)$.

3. Confusion matrix

   In the multi-class case, the confusion matrix is a square contingency matrix and it compares the true classes to the predicted classes. The diagonal indicates the number of instances per class which were correctly classified and the rest of the matrix indicates the number of misclassified instances, as well as what class they were predicted to be in.

   In the binary classification case, the two classes are viewed as a positive and a negative class and the confusion matrix presents the true positives, true negatives, false positives and false negatives. True positives and true negatives are correctly predicted instances for the positive and negative class respectively. False positives occur when the model predicts a negative instance to be positive and *vice versa* for false negatives. The confusion matrix provides a visual interpretation of which class is misclassified, especially in cases with class imbalance (Alpaydin, 2010).

4. Precision

   Precision indicates what proportion of positive predictions truly belong to the positive class. Precision is the ability of the model to avoid false positives or to avoid misclassifying negative instances as positive instances. Precision gives

information on how many predictions were correct (Harrington, 2012; Marsland, 2014).

5. Recall or sensitivity
   Recall indicates what proportion of the truly positive instances were correctly classified by the model as positive. Recall is the ability of the model to determine all positive instances. Recall gives information on how many true instances were misclassified (Harrington, 2012; Marsland, 2014).

6. F1-score, F-score or F-measure
   The F-measure calculates the weighted harmonic mean of precision and recall.

For the purposes of this research study, accuracy has been chosen as the classification performance metric, since it is a commonly used metric, is simplistic in nature and easy to understand.

### 4.4.3 Regression algorithms

The following regression algorithms will be covered by the study: DTs, KNR, LinReg, MLP, RFs and SVMs. DTs, KNRs, MLP, RFs and SVMs have been discussed in the previous section. These algorithms are most popularly used in practice and are well supported in *Python*. Table 4.9 presents the regression algorithms with the following characteristics for each: 1) the source where they can be researched in detail, including their mathematical formulations, 2) applications found in literature for a variety of data types, including time-series data and transactional data, and lastly 3) the application purposes as mentioned in Subsection 3.1.2.

#### 4.4.3.1 Linear regression

The goal of LinReg is to determine the relationship or function which maps inputs to outputs. Generally, this function is the sum of a deterministic function and some unexplained variability (Alpaydin, 2010). The unexplained variability is random noise. The deterministic function, also called 'regression equation', is a linear function. The function remains linear with an increase in input dimensionality; however, then it defines a linear hyperplane (Marsland, 2014).

Table 4.9: The table presents a summary of the applications of regression algorithms. Generally, regression is not applicable for text, image, audio and video data since these are mostly concerned with classification tasks.

| Algorithm | Source | Data Types | Applications |
|---|---|---|---|
| All | | | Regression |
| All DTS | | | Educational assistance<br>Reverse prediction<br>Simulation<br>Summation<br>Time-series prediction<br>Visualisation - tree diagram |
| CART | Alpaydin (2010)<br>Gupta *et al.* (2017)<br>Harrington (2012)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Time-series (Sen, 2018)<br>Transactional data (Timofeev, 2004) | |
| RF | Marsland (2014)<br>Pedregosa *et al.* (2011) | Time-series (Sen, 2018)<br>Transactional data (Liaw & Wiener, 2001) | |
| LinReg | LinearRegression | Textual data (Zhang & J. Oles, 2001)<br>Transactional data (Pedregosa *et al.*, 2011)<br>Time-series data (Sun *et al.*, 2003) | Summation<br>Simulation<br>Time-series prediction<br>Visualisation |
| All KNRs | | | Educational assistance<br>Simulation<br>Visualisation<br>Time-series prediction |
| KNR | Harrington (2012)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Transactional data (Alkhatib *et al.*, 2013)<br>Time-series data (Martínez *et al.*, 2017) | |

Continued on next page

| Algorithm | Source | Data Types | Applications |
|---|---|---|---|
| RNR | Pedregosa *et al.* (2011) | | |
| MLP | Alpaydin (2010)<br>Marsland (2014)<br>Pedregosa *et al.* (2011) | Transactional data (Paliwal & A. Kumar, 2009)<br>Time-series data (Koskela *et al.*, 1997) | Simulation<br>Time-series prediction |
| All SVCs | | | Data compression<br>Educational assistance<br>Simulation<br>Time-series prediction<br>Visualisation |
| Linear SVR | Alpaydin (2010)<br>Smola & Schölkopf (2004)<br>Chang & Lin (2011)<br>Pedregosa *et al.* (2011) | Image data(Basak *et al.*, 2007)<br>Time-series data (Sharma *et al.*, 2011)<br><br>Transactional data (Basak *et al.*, 2007) | |
| SVR with kernel | Alpaydin (2010)<br>Pedregosa *et al.* (2011) | Image data (Basak *et al.*, 2007)<br>Audio data (Basak *et al.*, 2007)<br>Time-series data (Sapankevych & Sankar, 2009; Sharma *et al.*, 2011)<br>Transactional data (Basak *et al.*, 2007) | |

The regression equation multiplies the regression weights by the feature values of an instance to determine the prediction for the instance. The process of determining these weights or parameters is called regression (Harrington, 2012). The sum of squared errors function (SSE) or the relative squared error (RSE) function is used to learn the weights through a process called the least-squares optimisation (Marsland, 2014). The derivative of the SSE error function (in terms of the weights) is determined and the set of parameters which minimises this function is called the least squares estimates or regression weights.

#### 4.4.3.2 The advantages and disadvantages of the selected regression algorithms

Table 4.10 presents the advantages and disadvantages of the selected purely regression algorithms, namely LinReg.

Table 4.10: Advantages and disadvantages of the selected purely regression algorithms, namely linear regression

| Advantages | Disadvantages |
|---|---|
| 1. The results are easily interpreted. | 1. It models non-linear data poorly. |
| 2. Is is a computationally inexpensive method (Harrington, 2012). | 2. It is prone to under-fitting the data. |
| 3. Multi-class classification is possible (Pedregosa *et al.*, 2011). | 3. It is only applicable for regression problems (Harrington, 2012) |
| | 4. It makes the assumption that the observations are independently and identically distributed, which makes it inappropriate for time-series data. |

#### 4.4.3.3 The different regression performance metrics

There are various metrics to evaluate the performance of regression algorithms, including explained variance score, mean absolute error, mean squared error, mean squared logarithmic error, median absolute error, the coefficient of determination, the root mean squared error, the mean or maximum relative error and the mean absolute percentage error. As mentioned in Section 4.3, regression applications have possible output options:

single output and multi-output. The above-mentioned metrics have been adjusted to handle the multi-output regression case as well. The mathematical formulations of the regression metrics are available in Pedregosa *et al.* (2011).

1. Explained variance score

   The explained variance measures the proportion to which the model represents the variation of the input dataset. The best possible score is 1. For multi-output regression, three options of presentation are available: an explained variance score per output, a single explained variance score which is the average of the outputs (uniform weighted), or an explained variance score which is the average of all outputs' scores which are weighted by the variances of each individual output.

2. Mean absolute error (MAE)

   The MAE compares the true output to the predicted output by calculating the average of the absolute difference between the predicted output and the true output. The MAE is a linear score, since the individual differences are weighted uniformly on the average. It summarises the model's performance independently from the direction of over-prediction or under-prediction. MAE is a risk metric which represents the expected value of the absolute error loss function. This metric is robust to outliers. The values are always non-negative and the best possible value is 0. For multi-output regression, two options of presentation are possible: an MAE per output or a single MAE which is the average of the uniform weighted outputs.

3. Mean squared error (MSE)

   The MSE compares the true output with the predicted output by calculating the variance of the differences between the predicted output and the true output. It is identical to calculating the average of the squared difference between the predicted output and the true output. It summarises the model's performance independently from the direction of over-prediction or under-prediction. It is a risk metric representing the expected value of the quadratic or squared error loss function. The values are always non-negative and the best possible value is 0. For multi-output regression, two options of presentation are available: an MSE per output or a single MSE which is the average of the uniform weighted outputs (Alpaydin, 2010).

4. Mean squared logarithmic error (MSLE)
   The MSLE is a risk metric representing the expected value of the squared logarithmic error loss function. Under-predictions are penalised more than over-predictions since this metric is more suitable for exponentially growing output, for example, population growth or sales of a commodity over a time-line. The values are always non-negative and the best possible value is 0. For multi-output regression, two options of presentation are possible: an MSLE per output or a single MSLE which is the average of the uniform weighted outputs.

5. Median absolute error (MedAE)
   The MedAE compares the true output with the predicted output by calculating the median error, where the error is calculated by subtracting the predicted value from the true value. This metric is robust to outliers. The value is always non-negative and the best possible value is 0. MedAE is not suitable for multi-output regression.

6. Coefficient of determination, $R^2$
   The $R^2$ measures how well the model fits the data. It indicates the proportion of the variance in the output which is predictable from the input. The value is in the range $(-1, 1)$ and the best possible value is 1. For multi-output regression, three options of presentation are available, namely a $R^2$ per output, a single $R^2$ which is the average of the uniform weighted $R^2$ per output, or a single $R^2$ which is the average of the $R^2$ of all outputs which are weighted by the variances of each individual output.

For the purposes of this research study, MSE has been chosen as the regression performance metric since it is since it is a commonly used metric, is simplistic in nature and easy to understand.

## 4.5 Conclusion: Chapter 4

This chapter introduced machine learning, the types and classes of machine learning algorithms and the selected machine learning algorithms for this research study. Each of the selected machine learning algorithms was discussed, including their applications

in literature, methodology, advantages and disadvantages, and the different metrics used to evaluate their performance.

In the following chapter, the decision support framework for this study will be conceptualised, developed and populated to present the final decision support framework of this study.

# Chapter 5

# Developing the decision support framework

The previous chapter discussed literature on machine learning (ML), including the definition of ML, the types of learning, the classes of ML and the different ML algorithms which will be used in this study.

In Chapters 3 and 4, Phases 1 to 5 of Jabareen's framework development methodology were applied. In this chapter, Phase 6 of Jabareen's methodology, 'Synthesise the concepts into a framework', will be applied. This chapter will present the conceptual framework developed. Thereafter, it will provide the implementation of the CRoss-Industry Standard Process for Data Mining (CRISP-DM) to aid in the design and implementations of the ML experiments to further the framework. This chapter will also present and discuss the datasets which will be used in this study, the preprocessing performed on them and the implementations of the presented ML algorithms. Thereafter, this chapter describes the development of the framework, by investigating the developed ML models. Lastly, this chapter presents the developed decision support framework. This chapter fulfils Objectives 2, 3 and 4 of this study.

## 5.1 Developing the conceptual framework

In this section the basic idea of the framework and the five criteria of the framework will be presented.

### 5.1.1 The basic idea for the decision support framework

As stated in Subsection 1.3, the goal of the research is to develop a decision support framework which considers both the *data characteristics* and the *application purpose* to indicate the appropriate *machine learning algorithm* for a given scenario. There are three main concepts: data characteristics, application purposes and ML algorithms. Together they form a three-dimensional framework, as illustrated in Figure 5.1. As per Phase 6 of Jabareen's framework development methodology, the identified concepts are used to construct the framework. Since it is difficult to illustrate a three-dimensional figure on a two-dimensional page, the framework will be cut along the $y$-axis and the interaction of the application purposes and ML algorithms will be illustrated per data type (each cut).

As stated in Subsection 3.2.1, data types or characteristics in this study refer to text, image, audio, video, time-series and transactional data. The application purposes for this research study were identified in Subsection 3.1.2, while the ML algorithms for this research study were identified in Section 4.4.

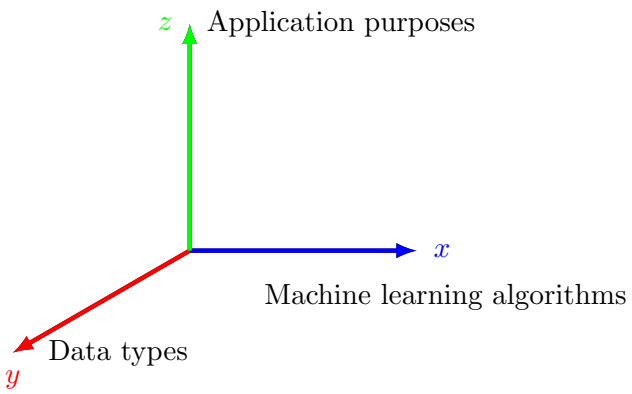


Figure 5.1: The basic idea for the decision support framework

### 5.1.2 The five criteria of the framework

As stated in Subsection 1.2, the decision support framework will be developed for a semi-skilled analyst, with mathematics, statistics and programming education, who is familiar with the process of programming, yet has not specialised in the variety of ML algorithms which are available. Different indicators will be needed to assist the analyst in understanding the factors involved per data type, application purpose and ML algorithm interaction. Therefore, the following five criteria have been developed to assist the analyst in the final choice of which ML algorithm to choose. The criteria are as follows:

1. Interpretability score per application purpose
   The interpretability score indicates the possibility to derive meaning from the results, given the user's or analyst's (human) capability. It is a binary score (Y or N) which describes whether the analyst can determine the relationship between the results of the application purpose and the data to make realistic sense of what is presented or discovered. This score is provided and explained per data type since the format or structure of the data influences its interpretability. It is based on the format of the preprocessed data. The score is also provided at the visualisation application purpose of the ML algorithms.

2. Programming requirements score per application purpose
   The programming score indicates the amount of programming which will be needed to achieve the desired application purpose. The different programming scores and their interpretations are presented in Table 5.1. In some cases the ML algorithm will provide the desired application purpose directly (dark green score) and no additional programming would be required. In other cases the analyst would have to convert the results of the ML algorithm into the desired results for the application purpose via programming. In the cases where the ML algorithms cannot accommodate an application purpose (not designed for it) and additional programming will not help, the programming score is white.

Table 5.1: The interpretation of the programming score

| | |
|---|---|
| Given by the algorithm; no programming needed | |
| Mostly given by the algorithm/package; little programming needed | |
| Little given by the algorithm; program the rest | |
| Full programming, by using only the outputted labels, is required | |
| Not available | |

3. Recommendation score per application purpose

   The recommendation score indicates the usefulness of the result of the programming task given the data properties or format and the ability to replicate the developed model.

   The recommendation score is a numerical value which is provided per programming score. The different recommendation scores and their interpretations are presented in Table 5.2. This score will not be indicated at dark green and white programming scores, since dark green indicates the model provides the desired application directly (no additional programming required) and white indicates that the model does not provide the desired application purpose (it is not available). The recommendation score is linked to the light green, yellow and orange programming scores.

   This score is not provided for the visualisation application purpose. Visualisation does not convert the results of the ML algorithm; it only illustrates it. The recommendation score is also not provided for the educational assistance application purpose since it can be determined what instances in the dataset to inspect, but not how to inspect them in order to derive meaningful results. That depends on the dataset itself and a subject matter expert (SME) will be required.

4. Performance score per machine learning algorithm

   The performance score of the ML algorithm is given as a ranking score in ascending order (lower value indicates better performance and *vice versa*). The Fowlkes-Mallows index/score (FMI) will be used for clustering algorithms, the accuracy score for classification algorithms and the mean squared error (MSE) for regression algorithms. The best score is highlighted in green, the worst in red and the average performers in orange.

Table 5.2: The interpretation of the recommendation score

| | |
|---|---|
| Exact copy of the model is created | 3 |
| Serves as a guideline | 2 |
| Do not use | 0 |

5. Execution time score per machine learning algorithm

   The execution time is the time it took the ML algorithm to execute the full implementation, given as a ranking score in ascending order (lower value indicates less execution time and *vice versa*).

   For classification and regression algorithms the execution time is measured from fitting the training dataset to predicting the output of the validation dataset. In the case of clustering algorithms, only the time to fit the data is used. The best score is highlighted in green, the worst in red and the average performers in orange.

## 5.2 Populating the conceptual framework I

In this section the development of the conceptual framework will be further described. This section will focus on evaluating the performance and execution time scores of the ML algorithms, by applying Phases 3 and 4 of the CRISP-DM to design and implement the experimentations with the different ML algorithms. This section will present the datasets used in this study, the data preprocessing and the ML algorithm implementations. Lastly, each ML algorithm implementation will be investigated and evaluated according to the criteria of the framework.

### 5.2.1 The datasets

Six different data types were used in this research study: text, image, audio, video, transactional and time-series data. All six data types are applicable to clustering and classification; however, only transactional and time-series data are applicable to regression. The datasets for classification are also applicable for clustering, since both types of algorithms are concerned with the grouping of instances into classes/clusters. A total of 75 datasets were collected from online, public data repositories, including

Kaggle and UCI. Summaries of the datasets, including their sources and references are available in Appendix A.

### 5.2.2 Data preparation and preprocessing

The following data preprocessing methods, as specified by the third phase of the CRISP-DM, 'Data preparation and preprocessing', were performed on the collected datasets:

#### 5.2.2.1 Data cleaning

The datasets should be examined for outliers, errors and missing values and treated according to the strategies and methods provided in Subsection 3.3.1. The outliers were not removed, since the ML algorithms were evaluated on their ability to recognise outliers. Errors in terms of inconsistent values compared to the rest of the features, for example, text in a numerical valued feature, were removed as if they were missing values. Missing values were treated using the discard strategy as mentioned in Subsection 3.3.1.3, since the imputing strategy would be a time-consuming process. The researcher also could not determine the correct values since she does not have access to the infrastructures which created the datasets.

#### 5.2.2.2 Data transformation

The numerical variables were transformed according to the zero-to-one standardisation, since the researcher wanted to preserve the original data distributions as far as possible. Using the $z$-score would introduce an underlying distribution to the data. Not all ML algorithms can handle negative values; thus, the $[-1, 1]$ value range was avoided. The zero-to-one standardisation also enables the testing of the capabilities of the ML algorithms to perform outlier detection as an application purpose since it emphasises outliers.

The categorical variables were replaced by the binary dummy variables as explained in Subsection 3.3.2.2 and illustrated in Table 3.1.

For clustering and classification, the target variables were transformed to integers starting at 0. For regression the target variables were transformed to the $[0, 1]$ value range using the zero-to-one standardisation.

#### 5.2.2.3 Normalisation

Normalisation is not applicable for ML algorithms, since the ML algorithms perform it themselves, as explained in Subsection 3.3.3.

#### 5.2.2.4 Filtering

Unnecessary or insignificant features were filtered or removed from the datasets where needed. For example, in the parking time-series dataset both the size and current occupancy of each parking area were provided per time stamp, whereas only the occupancy over time is needed. The size remained constant throughout the time stamps and was removed from the dataset since it did not provide any additional information.

#### 5.2.2.5 Abstraction

Temporal and qualitative abstraction was employed. Temporal abstraction was used in the ordering of audio data, image data and time-series data to ensure that the order of the instances were kept and correctly represented. Qualitative abstraction was used to map variables to a better representative format, for example, 'Monday' was mapped to a value of '1' and 'Friday' to a value of '5'. Another example is the mapping of '30 minutes to 1 hour dedicated to homework' to a value of '1' and '1 to 2 hours dedicated to homework' to a value of '2'.

#### 5.2.2.6 Reduction

As stated in the research scope and assumptions in Section 1.7, reduction techniques are not included in this research study.

#### 5.2.2.7 Derivation

Target variable derivation was used to transform the targets suitable for regression into targets suitable for classification. For example, with the stock time-series dataset the stock price was transformed into '0's and '1's where a '0' indicates a decrease in stock price between the current and the previous time stamps and a '1' indicates an increase.

#### 5.2.2.8 Data division

The datasets were split in a 70:30 percentage ratio for training and testing datasets respectively, unless otherwise stated by the data source. Training datasets are not required for the research study since there will not be any parameter tuning of the developed models.

### 5.2.3 Data type specific preparation and preprocessing

Each data type requires its own preprocessing to transform it into a suitable format for the ML algorithms. All the datasets were shuffled after the data type specific preprocessing, except for the time-series datasets, where the order of instances should be retained.

#### 5.2.3.1 Text data

The 10 text datasets were converted from strings of characters to the applicable input format for the ML algorithms by using the *CountVectorizer* and *TfidfTransformer* packages of *sklearn* in *Python*. These packages transform the text data into sparse matrices containing numerical values.

#### 5.2.3.2 Image data

A total of 11 image datasets were used. Images consist of pixels and their dimension is presented as the number of pixels in the width by the number of pixels in the height of the image. Pixels are represented by numerical values. In the case of greyscale, each pixel is represented by a single value. In the case of a colour image, each pixel is represented by four values: the three RGB values (Red, Green and Blue) and the fourth is the transparency value. The images were converted to one-dimensional vectors of pixel values before applying the ML algorithms.

#### 5.2.3.3 Audio data

A total of 10 audio datasets were used. Audio files consist of different features which change over time, meaning there is a time component. Eight audio features were extracted from each of the audio files at a constant sampling frequency. Each audio file was represented by a two-dimensional array where each column represented one of

the eight audio features and each row represented a successive sample taken over time. The order of the rows per audio file was kept intact throughout the ML process.

#### 5.2.3.4 Video data

A total of 10 video datasets were used. For this research silent videos were utilised. Each video file consists of a sequence of images over time, meaning there is a time component. Each video was converted into a set of successive images and the images were converted into one-dimensional pixel vectors. The order of images was kept intact per video set throughout the ML process.

#### 5.2.3.5 Transactional data

A total of 14 transactional datasets for clustering/classification and 15 transactional datasets for regression were used. The transactional datasets did not require additional special preprocessing.

#### 5.2.3.6 Time-series data

A total of 10 time-series datasets for clustering/classification and regression respectively, were used. The time-series datasets used in this research study had one of two possible formats: the dataset consists of either successive instances or subsets of successive instances. An example of successive instances is the sales of bikes for a year (365 instances). An example of subsets of successive instances is the sales of 20 different products in a year ($(20 \times 365)$ instances).

In terms of the data cleaning of time-series data, data snippets or successive instances were chosen such that the order was uninterrupted by missing entries. The time-series data was split such that the training dataset is the past sequence of instances and the validation set is the sequential set of instances to the training dataset.

### 5.2.4 Building and implementing the models

This section presents the different ML algorithms and their implementations in *Python*, as per Phase 4 of the CRISP-DM. It also discusses the problems encountered during the preliminary deployment of the models and the steps taken to address the problems.

#### 5.2.4.1 The machine learning algorithm implementations

Table 5.3 presents each ML algorithm, the *Python* package to implement it and its parameter specifications as implemented in *Python*. The column '#' indicates the number of variations of the algorithm. The parameter specifications were kept as recommended by *Python* unless otherwise stated. The package names in the table form hyperlinks to the *Python* package documentation and the analyst can investigate it to determine all the parameters available, including parameters for tuning.

Herewith are some important notes on the implementations of the ML algorithms:

1. All preprocessing, implementations and experiments were performed in *Python* using the *sklearn* libraries.

2. The algorithms were implemented in a sequential fashion and only one algorithm was executed at a time.

3. Four identical computers were utilised: 12 GB RAM, i7 core, *Windows 10*.

4. During the implementations of the ML algorithms, no parameters were specified (except for the MLP) or experimentally tuned. The goal of the study is to provide a guideline of the applicable algorithms in terms of the application purposes and data types of the analysts, and to provide a starting point from which to choose an appropriate ML algorithm.

5. The datasets were shuffled, except for the time-series datasets, where the order of instances should be retained.

6. For each data characteristic, the associated algorithms were implemented totalling 14 scripts (six for clustering, six for classification and two for regression).

Table 5.3: The implementations of the machine learning algorithms in *Python*

| Algorithm | *Python* | # | Parameters |
|---|---|---|---|
| **Clustering** | | | |
| AHC | AgglomerativeClustering | 4 | 4 linkages: average, complete, single and ward<br>Number of clusters |
| DBSCAN | DBSCAN | 3 | 3 solvers: ball-tree, brute and KD-tree |
| KMC | KMeans | 1 | Number of clusters |
| miniKMC | MiniBatchKMeans | 1 | Number of clusters |
| MS | MeanShift | 1 | |
| One-class SVM | OneClassSVM | 4 | 4 kernel functions: linear (LIN), polynomial (POLY),<br>radial basis function (RBF) and sigmoid (SIG) |
| | | 14 | |
| **Classification** | | | |
| CART | DecisionTreeClassifier | 1 | |
| ET | ExtraTreeClassifier | 1 | |
| EF | ExtraTreesClassifier | 2 | Forest with 10 and 100 trees respectively |
| ID3 | Id3Estimator | 1 | |
| RF | RandomForestClassifier | 2 | Forest with 10 and 100 trees respectively |
| KNN | KNeighborsClassifier | 4 | 4 Solvers: auto, ball-tree, brute and KD-tree |
| RNN | RadiusNeighborsClassifier | 4 | 4 Solvers: auto, ball-tree, brute and KD-tree |
| NC | NearestCentroid | 1 | |
| LogReg | LogisticRegression | 5 | 5 solvers: LBFGS, LibLinear (LIBLIN), Newton-CG (NTCG),<br>SAG and SAGA |
| MLP | MLPClassifier | 12 | 4 activation functions: identity, logistic, ReLu and tanh<br>3 Solvers per activation function: ADAM, LBFGS and SGD<br>Number of hidden layers: 15<br>Learning rate = 0.01<br>Maximum iterations: 500 |

Continued on next page

| Algorithm | *Python* | # | Parameters |
|---|---|---|---|
| NB | Naïve Bayes | 4 | BNB, CNB GNB and MNB |
| SVC | SVC | 4 | 4 kernel functions: LIN, POLY, RBF and SIG |
| $\upsilon$-SVC | NuSVC | 4 | 4 kernel functions: LIN, POLY, RBF and SIG |
| | | 45 | |
| **Regression** | | | |
| CART | DecisionTreeRegressor | 1 | |
| ET | ExtraTreeRegressor | 1 | |
| EF | ExtraTreesRegressor | 2 | Forest with 10 and 100 trees respectively |
| ID3 | Id3Estimator | 1 | |
| RF | RandomForestRegressor | 2 | Forest with 10 and 100 trees respectively |
| KNR | KNeighborsRegressor | 4 | 4 Solvers: auto, ball-tree, brute and KD-tree |
| RNR | RadiusNeighborsRegressor | 4 | 4 Solvers: auto, ball-tree, brute and KD-tree |
| LinReg | LinearRegression | | |
| MLP | MLPRegressor | 12 | 4 activation functions: identity, logistic, ReLu and tanh<br>3 Solvers per activation function: ADAM, LBFGS and SGD<br>Number of hidden layers: 15<br>Learning rate = 0.01<br>Maximum iterations: 500 |
| SVR | SVR | 4 | 4 kernel functions: LIN, POLY, RBF and SIG |
| $\upsilon$-SVR | NuSVR | 4 | 4 kernel functions: LIN, POLY, RBF and SIG |
| | | 35 | |

### 5.2.5 Problems encountered during the preliminary model deployment

Two types of problems were encountered during the preliminary deployment of the ML algorithms: problematic ML algorithms and memory errors.

#### 5.2.5.1 Problematic machine learning algorithms

Algorithms were named problematic when their performance was inconsistent across all data types when their parameters were kept at fixed values. These algorithms provided results for some datasets, instead of for all the datasets. To solve this problem, parameter tuning is required per data type and dataset, which defeats the goal of the study. Therefore, these algorithms were removed from the list of evaluated ML algorithms.

The problematic ML algorithms for classification were RNN and NuSVC and for regression it was RNR.

#### 5.2.5.2 Memory errors

Memory errors occurred when the datasets were too large considering the computational power available *vs* the computational power required by the ML algorithms. The memory errors were addressed by implementing the following solutions.

1. For text datasets, the preprocessed datasets were limited to 50 million entries (number of rows $\times$ number of columns) to avoid memory errors.

2. For clustering algorithms the number of instances was limited to 40 000 instances and for classification and regression algorithms the number of instances was limited to 100 000 instances to avoid memory errors. With the reduced datasets, the original ratio of classes (class balances) was preserved to better represent the original datasets.

3. An error occurred during the experimental runs of the audio datasets with the SVCs. The dataset sizes were changed to approximately 100 000, 90 000, 80 000 and 70 000 instances; however, the algorithms did not converge. They were terminated when no results were yielded after 24 hours (per individual algorithm).

Thus, SVCs were deemed inappropriate for audio data and the associated five criteria of the framework were not applied to them.

### 5.2.6 Evaluating the performance and execution time scores

After the preliminary problems were addressed, the ML algorithms were fully deployed. Each ML algorithm was implemented six times per data type and dataset, since some ML algorithms depend on a measure of randomness which influences their results.

The averages of the performance measurements (FMI for clustering, accuracy for classification and MSE for regression) were calculated and ranked in ascending order. Both the average values and ranks were added to the frameworks to provide an indication of what performance was achieved on average by the ML algorithms (values) and which ML algorithms performed best, average and worst (ranks).

The averages of the execution times (in seconds) were calculated and ranked in ascending order. Both the average values and ranks were added to the frameworks to provide an indication of what average execution time was achieved by the ML algorithms (values) and which ML algorithms performed best, average and worst (ranks).

The performance and execution time scores were added to the $z$-axis of the framework in Figure 5.1, below the application purposes, since it is indicated per ML algorithm which is on the $x$-axis.

## 5.3 Populating the conceptual framework II

This section will focus on evaluating the interpretability, programming and recommendation scores, by investigating the developed models. As previously stated, SVCs were deemed inappropriate for audio data and the five criteria of the framework were not applied to them.

### 5.3.1 Evaluating the programming scores

In this section the programming score of each application purpose and ML algorithm pair ($(z, x)$ pairs on Figure 5.1) is explained and provided in Table 5.4. The results are consistent regardless of the data type used since it depends on the outputs the ML algorithm provides.

A note on manual programming at clustering and classification:

One can manually program the association, data compression and outlier detection application purposes. One can use the predictions of the algorithm to group the instances, calculate the mean of each cluster/class and define a radius distance in the range $(0, 1)$ to get ranges per feature, which creates "if-then" rules for clustering/classifying the data. The score colour depends on what is provided by the algorithm. If only predictions are provided, the score would be orange. If the cluster/class centres are provided by the algorithm, then the score would be yellow since less programming would be needed to achieve this result. The "if-then" rules can be used for association. The centres and radius distances can be used for data compression, since only the centres and radius distances per feature would be needed to make the prediction for a new data point. It can also be used for outlier detection since instances outside the feature ranges or radii can be identified as outliers.

A note on manual programming at regression:

One can manually program the outlier detection, optimisation and reverse prediction application purposes. For outlier detection, one can plot the $R^2$ (coefficient of determination) plot and define a margin in the range $(0, 1)$ on both sides of the regression (ground truth) line. The data points which fall in the (triangular) regions outside of this double margin can be labelled as outliers.

For optimisation one can program to search the dataset for the minimum or maximum target values (depending on the definition of optimisation for the particular dataset, for example, maximum profit or minimum cost). In the case of categorical data, one can use the majority vote per feature to determine the dependent variables which lead to the optimised target or result. In the case of numerical data, one can use the averages per feature to determine the dependent variables which lead to the optimised target or result.

For reverse prediction one can program to search the dataset for the desired target values and use the same methods as for the manual optimisation to determine the dependent variables which lead to the desired target or result. The programming score for these three manually programmable options is orange since no additional outputs, apart from the predictions, of the ML algorithms are required.

Table 5.4: The programming score per application purpose and machine learning algorithm pair

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Association** | | | |
| Clustering algorithms | AHC | It has two manual programming options. The first is to use the predictions and perform manual association which would be a bottom-down process. The second option is to work from the bottom upwards to determine the "if-then" rules for similarity to add groups together (try to mimic the algorithm). The latter is done by using the 'children' provided by the algorithm, which indicates in a step-wise manner which instances were grouped to form the next group and then the next group and so forth. The second option would require much programming, time-investment and would lead to many "if-then" rules with many overlapping areas. It is also doubtful whether it would deliver a good representation of the model. Therefore, the first option is recommended. | |
| | DBSCAN | It additionally provides the core data points and by using only the predictions of the core samples in manual association, the accuracy of the association result is improved. The score is yellow since additional information from the algorithm output is used in the programming process. | |
| | KMC and miniKMC | They additionally provide the centroids of the clusters (averages of instances in the clusters identified by the algorithm) and manual association is used. | |
| | MS | It additionally provides the centroids of the clusters and manual association is used. | |
| | One-class SVM | It does not provide additional information and manual association is used. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Association** (cont.) Classification algorithms | DTs | They have two options. The first is to use the add-on package (thus, the score is light green) to create a decision tree diagram, which visually illustrates the decisions and conditions followed by the model until a leaf (end) node is reached. Next, "if-then" rules can be derived from it. The second option is to perform manual association. The first option is better since it accurately copies the model. | |
| | KNN | It does not provide additional information and manual association is used. | |
| | NC | It additionally provides the centroids of the classes and manual association is used. | |
| | LogReg | It does not provide additional information and manual association is used. | |
| | MLP | It does not provide additional information and manual association is used. | |
| | NB | The GNB algorithm additionally provides the centroids of the classes and manual association is used. The rest of the NB algorithms do not provide additional information and manual association is implemented. | |
| | SVC | It does not provide additional information and manual association is used. | |
| Regression algorithms | DTs | They have the decision tree diagram package (refer to the detail at classification DTs) which can be used to derive the "if-then" rules. Since regression works with continuous valued labels, it would most likely be that there are many "if-then" rules to accurately capture all the labels. | |
| | The rest | For the rest of the regression algorithms manually programming the "if-then" rules would be too complex a task. Also, the algorithms do not provide any supplementary information to aid in this task. | |
| **Classification** Clustering algorithms | All | They cannot perform classification. | |
| Classification algorithms | All | They can perform classification. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Classification** (cont.) Regression algorithms | All | They cannot perform classification. | |
| **Clustering** Clustering algorithms | All | They can perform clustering. | |
| Classification algorithms | All | They cannot perform clustering. | |
| Regression algorithms | All | They cannot perform clustering. | |
| **Data compression** Clustering algorithms | AHC | It has two manual programming options. The first is to perform manual data compression (a bottom-down process). The second option is to work from the bottom upwards to determine the "if-then" rules for similarity to add groups together (try to mimic the algorithm). The last option would require much programming, time-investment and would lead to many "if-then" rules with many overlapping areas. It is doubtful whether it would perform data compression since there will be many "if-then" rules which are created by going stepwise through the model's operations. Therefore, the first option is recommended. | |
| | DBSCAN | It additionally indicates the core data points and by using only the predictions of the core samples in manual data compression, the accuracy of the result is improved. The score is yellow since additional information is used. | |
| | KMC and miniKMC | They additionally provide the centroids of the clusters. For both algorithms only the centres are used to make predictions with, since the centre closest to the new instance indicates its cluster. Only centres need be stored for data compression. | |
| | MS | It additionally provides the centroids of the cluster and the same reasoning as for KMC holds. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Data compression** (cont.) | One-class SVM | It additionally provides the support vectors, weights (coefficients) and the bias (intercept) used in the decision function, which can be used to achieve exactly the same results as the model. | |
| Clustering algorithms (cont.) | | In the case of the LIN kernel, the score is light green since a simple multiplication operation is required. | |
| | | In the case of the other three kernels, the kernel parameters have to be known (for example, the polynomial degree) and the kernel functions have to be manually programmed. The score is yellow, since it requires programming effort while using additional output of the algorithm. | |
| Classification algorithms | DTs | They have two options. The first is to use the decision tree diagram package and store the "if-then" rules. The second option is to perform manual data compression. The first option is better since it accurately copies the model. It can also be combined with the feature importances (explained at the summation application purpose) to only retain the features which are relevant. | |
| | KNN | It does not provide additional information and manual data compression is used. | |
| | NC | It additionally provides the centroids of the classes. For NC only the centres are used to make predictions with, since the centre closest to the new instance indicates its class. Only centres need be stored for data compression. | |
| | LogReg | It additionally provides the coefficients and the intercept used in the decision function, which can be used to achieve exactly the same results as the model. The score is light green since a simple multiplication operation is required. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Data compression** (cont.) Classification algorithms (cont.) | MLP | It additionally provides the coefficients and the intercepts used in the decision function (input and hidden layers). A new instance or data point is multiplied by the coefficients of the input layer and the associated intercept is added. The result is fed to the activation function (needs to be programmed manually) and that result is then multiplied by the coefficients of the hidden layer and the associated intercept is added. This is the output of the hidden layer and since it is a classification task, the result is fed to the soft-max function (needs to be manually programmed) to determine the class of the instance. The score is yellow, since it requires programming effort while using additional output of the algorithm. | |
| | NB: | The GNB algorithm additionally provides the centroids of the classes and manual data compression is used. | |
| | | The rest of the NB algorithms do not provide additional information and manual data compression is used | |
| | SVC | It does not provide useful additional information and manual data compression is used. Due to the implementation of the algorithm in *Python* the decision boundary cannot be replicated with the use of the support vectors, coefficients and intercepts which are provided by the algorithm. | |
| Regression algorithms | DTs | They can use the decision tree diagram package and store the "if-then" rules. It can also be combined with the feature importances (explained at the summation application purpose) to only retain the features which are relevant. | |
| | KNR | It does not provide a data compression component. | |
| | LinReg | It additionally provides the coefficients and the intercept used in the decision function, which can be used to achieve exactly the same results as the model. The score is light green since a simple multiplication operation is required. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Data compression** (cont.) Regression algorithms (cont.) | MLP | It additionally provides the coefficients and the intercepts used in the decision function. The same process as for the MLP at classification holds, with one change: the output of the hidden layer is the prediction of the model (soft-max function is not used for regression) | |
| | SVR | It additionally provides the support vectors, weights (coefficients) and the bias (intercept) used in the decision function, which can be used to achieve exactly the same results as the model. The same process as for the one-class SVM at clustering holds. | |
| | | In the case of the LIN kernel | |
| | | In the case of the other three kernels | |
| **Educational assistance** | | | |
| Clustering algorithms | AHC | It has an add-on package to create a dendrogram, which visually illustrates how the model groups instances from the bottom upwards. Correlations and similarities between clusters can be seen and possible further split points in the data or clusters can be indicated. Further inspection of the 'children' instances will be needed; thus, more programming is needed. | |
| | The rest | They do not provide an educational component. | |
| Classification and Regression algorithms | DT | The "if-then" rules derived from the decision tree diagram can be inspected to determine if any information can be learnt. The DTs (except the ID3) provide a sparse matrix which indicates per instance the nodes (in the tree) it followed until it reached a leaf node. This, together with the "if-then" rules can be inspected. This will require more programming | |
| | The rest | They do not provide an educational component. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Optimisation** | | | |
| Clustering and Classification algorithms | All | According to the definition of optimisation, this application purpose desires the dependent variables of the best instance. This application purpose is unavailable for clustering or classification, since with clusters/classes there is not a best, there are only clusters/classes. Also, if one would like to work back from a cluster/class to get the values of the features, no specific values will be derived but rather ranges or multiple values for the features. Therefore, a single solution is not possible. | |
| Regression algorithms | All | None of the regression algorithms provide this application purpose directly. Manual optimisation can be used instead. | |
| **Outlier detection** | | | |
| Clustering algorithms | AHC | It has two manual programming options. The first option is to perform manual outlier detection (a bottom-down process). The second option is to work from the bottom upwards to determine the "if-then" rules for similarity to add groups together (as explained above). It is doubtful whether it would enable outlier detection since there will be many "if-then" rules with overlap which are created by going stepwise through the model's operations. The overlap leads to reduced outlier detection capability. Therefore, the first option is recommended. | |
| | DBSCAN | It provides the outliers, which are given labels of -1. | |
| | KMC and miniKMC | For KMC and miniKMC no outlier detection is possible since a new instance is assigned to the cluster of the cluster centre closest to it. Manual outlier detection can be used instead and the algorithms additionally provide the cluster centres. | |
| | MS | For MS the same reasoning as for KMC holds. | |
| | One-class SVM | It is designed for outlier detection. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Outlier detection** (cont.) | DTs | They do not perform outlier detection since the decision tree rules cover the entire decision space. Manual outlier detection can be used instead. | |
| Classification algorithms | KNN | It does not provide additional information and manual outlier detection is used. | |
| | NC | For NC no outlier detection is possible since a new instance is assigned to the class of the class centre closest to it. Manual outlier detection can be used instead. The algorithm additionally provides the cluster centres. | |
| | LogReg | It does not provide additional information and manual outlier detection is used. | |
| | MLP | It does not provide additional information and manual outlier detection is used. | |
| | NB | The GNB algorithm additionally provides the centroids of the classes and manual outlier detection is used. | |
| | | The rest of the NB algorithms do not provide additional information and manual outlier detection is used. | . |
| | SVC | It does not provide additional information and manual outlier detection is used. | |
| Regression algorithms | DT | They have two options. The first is to use the decision tree diagram package; however, then outlier detection cannot be performed since the decision tree rules cover the entire decision space (programming score would be white). The second option is to perform manual outlier detection. The second option is chosen since it provides outliers. | |
| | The rest | They do not provide this application purpose directly and manual outlier detection is used instead. | |
| **Regression** Clustering algorithms | All | They cannot perform regression. | |
| Classification algorithms | All | They cannot perform regression. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Regression** (cont.) | | | |
| Regression algorithms | All | They can perform regression. | |
| **Reverse prediction** | | | |
| Clustering and Classification algorithms | All | According to the definition of reverse prediction, this application purpose desires the dependent variables of a single desired instance. This application purpose is unavailable for clustering or classification, since with clusters/classes there are no individual instances, there are only clusters/classes. Also, if one would like to work back from a cluster or class to get the values of the features, no specific values will be derived but rather ranges or multiple values for the features. Therefore, a single solution is not possible. | |
| Regression algorithms | All | None of the regression algorithms provide this application purpose directly. Manual reverse prediction can be used instead. | |
| **Simulations** | | | |
| Clustering algorithms | AHC and DBSCAN | These algorithms cannot perform simulations of single instances since they do not store the developed model. | |
| | The rest | They store the developed model and predictions of single instances are possible. | |
| Classification and Regression algorithms | All | They can perform simulations since they store the developed model and predictions of single instances are possible. | |
| **Summation** | | | |
| Clustering algorithms | All | None of the clustering algorithms provide additional information of what they have discovered or learnt from the data. | |

Continued on next page

| Application $z$-axis | Algorithm $x$-axis | Investigation | Score |
|---|---|---|---|
| **Summation** (cont.) Classification and Regression algorithms | DTs | The ID3 tree does not provide any summations on the data and the model; however, the rest of the DTs do. A feature importance measurement is provided which indicates the importance or influence of a feature. It indicates the relative contribution of each feature to the output or predictions of the ML algorithm. It is a percentage value and the sum of all feature importances is 1. | |
| | The rest | They do not provide a summation component. | |
| **Transparency** | | | |
| Clustering algorithms | AHC | It provides the 'children', which indicates in a stepwise manner which instances were grouped to form the next group and then the next group and so forth. | |
| | The rest | They do not provide a transparency component. | |
| Classification and Regression | DTs | The ID3 tree does not provide any transparency on the data and the model; however, the rest of the DTs do. A sparse matrix is provided which indicates per instance the nodes (in the tree) it followed until it reached a leaf node. | |
| | KNN and KNR | They provide the neighbours used to determine the prediction of an instance as well as the distances of the neighbours to the instance. | |
| | The rest | They do not provide a transparency component. | |
| **Visualisation** | | | |
| Clustering algorithms | AHC | It provides the dendrogram. | |
| | The rest | Visualisation can be done by manually programming the plots or graphs using the data and the predictions outputted by the algorithms. | |
| Classification and Regression algorithms | DTs | The decision tree diagram visually illustrates the decisions and conditions used in the decision making process of the model. The conditions are in the form of the data feature values. | |
| | The rest | Visualisation can be done by manually programming the plots or graphs using the data and the labels outputted by the algorithms. | |

### 5.3.2 Evaluating the interpretability and recommendation scores

In this section the interpretability and recommendation scores are explained per data type. As previously stated, SVCs were deemed inappropriate for audio data and the five criteria of the framework were not applied to them.

#### 5.3.2.1 The text data

The interpretability score for the data type as a whole is N since the data is transformed to a sparse matrix. The correlation between the features and the values of the features are unknown and not interpretable.

In terms of clustering, the interpretability score for the visualisation application purpose is Y only for AHC since a dendrogram is interpretable. For the rest of the clustering algorithms it is not applicable. In terms of classification, the interpretability score for the visualisation application purpose is N for the DTs since a decision tree diagram is a visualisation option; however, it is not interpretable since the feature values and the values of the "if-then" rules are in terms of the sparse matrix. For the rest of the classification algorithms it is not applicable.

In terms of the programming score, the only deviation from the general rule is the visualisation of the data. Plotting the data to graphs is not possible since the data is sparse.

The recommendation score is mostly 0, except in the cases where the result is an exact copy of the learnt model. Since the input data is a sparse matrix, the centres of the features can be calculated; however, since there are few instances per feature, defining a radius might be impractical since no other instances might be captured. In that case many instances would be falsely labelled as outliers. Therefore, the manual programming of association, data compression and outlier detection is not recommended.

#### 5.3.2.2 Image data

The interpretability score for the data type as a whole is N.

The clustering interpretability score for the visualisation application purpose is Y for AHC (dendrogram) and N for the rest of the algorithms. The classification interpretability score for the visualisation application purpose is N. The data is pixelated and every image is parsed to a one-dimensional vector; thus, each feature represents a

pixel location. An image cannot be clustered on a singular pixel but a group of pixels; thus, plotting two features on a two-dimensional plot lacks information to draw a distinction between them. In different images of, for example, the letter 'c', the letter could be at different locations on the image (left bottom corner or right top corner) or the scale per image could be different (an upper case 'C' or a lower case 'c').

This inconsistency indicates that for the same cluster (the cluster for 'c') the pixel values of a single location (feature) can vary between the whole range of pixel values and therefore the distribution is distorted. There will also be much overlap in pixel values for different clusters, for example, a '0', '8' and '6' (on the same scale and location) share many pixel locations which will overlap in a plot, as illustrated in Figure 5.2, and no distinction can be made between them. They will also share feature centres as calculated in the manual programming of the centres.

Subsequently, the recommendation score is mostly 0, except in the cases where the result is an exact copy of the learnt model.

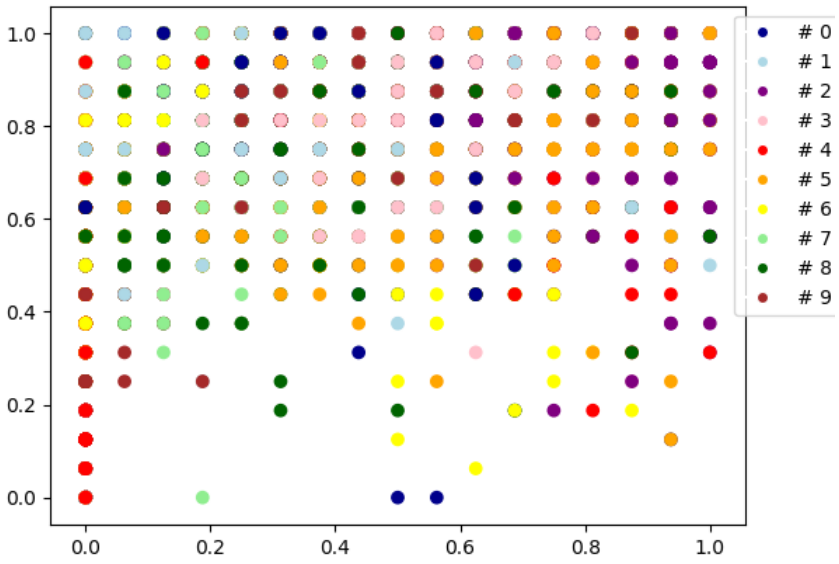


Figure 5.2: The results when plotting image data. There are 6 520 instances in the digit dataset; however, due to overlapping only 210 instances are visualised. The digit dataset consists of handwritten digits and each colour on the image represent a class or a value in the integer range of zero to nine.

#### 5.3.2.3 Audio data

The interpretability score for the data type as a whole and interpretability score for the visualisation application purpose for both clustering and classification is Y since the correlations between the features and the values of the features are known and interpretable as illustrated in Figure 5.3, where two audio features are plotted, namely short time energy *vs* spectrum flux. It seems that dogs tend to have higher short time energy and lower spectrum flux values compared to cats. It is possible to derive realistic sense from the data. Subsequently, the recommendation score is 2, unless the result is an exact copy of the learnt model (score is 3).

#### 5.3.2.4 Video data

The interpretability score for the data type as a whole is N. The clustering interpretability score for the visualisation application purpose is Y for AHC (dendrogram) and N for the rest of the clustering algorithms. The classification interpretability score for the visualisation application purpose is N. The same explanation as for image data in Subsection 5.3.2.2 is valid here for the interpretability and recommendation scores, since the videos are sequences of images, each of which was also parsed to one-dimensional vectors.

#### 5.3.2.5 Transactional data

Transactional data can consist of categorical data, numerical data or a mixture of both.

1. Pure categorical data

   The overall interpretability score is Y since correlation between the features and the values of the features are known and interpretable. The clustering interpretability score for the visualisation application purpose is Y for AHC and the classification and regression interpretability score for the visualisation application purpose is Y for DTs. For the rest of the clustering, classification and regression algorithms, the score is N since the categorical values are either 0 or 1; thus, only four data points ( (0,0), (0,1), (1,1), (1,0) ) can be plotted when plotting features. This results in overlap of the data points and no distinction is possible, as illustrated in Figure 5.4.

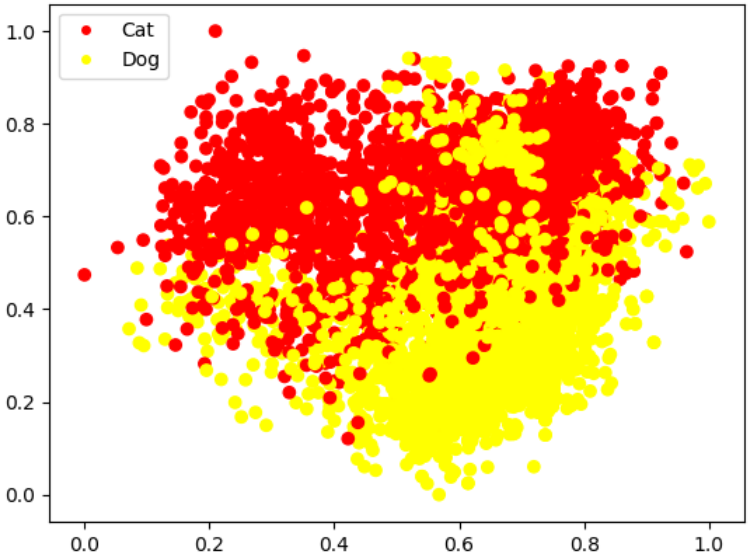


Figure 5.3: The results when plotting two audio features, short time energy (vertical axis) *vs* spectrum flux (horizontal axis) for an audio dataset for cats and dogs

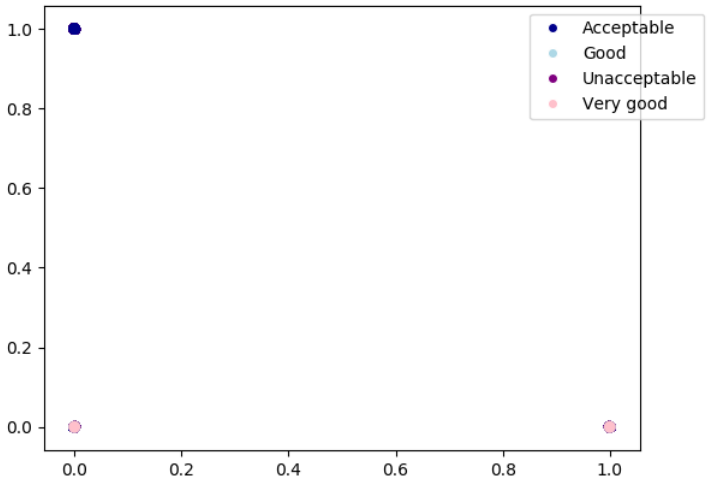


Figure 5.4: The results when plotting categorical data. There are 1 728 instances in the car evaluation dataset; however, due to overlapping only three instances are visualised.

For clustering and classification, the recommendation score is mostly 0, except in the cases where the result is an exact copy of the learnt model. Since the input data is 0s and 1s, the centres of the features can be calculated; however, the values would be averages and located far from the actual data points. Defining a radius is impractical since few other instances will be captured. In that case, many instances would be falsely labelled as outliers. Therefore, the manual programming of association, data compression and outlier detection is not recommended.

For regression, the recommendation score is 2, unless the result is an exact copy of the learnt model (score is 3).

2. Pure numerical data
   The overall interpretability score and interpretability scores for the visualisation application purpose is Y since correlation between the features and the values of the features are known and interpretable. The recommendation score is 2, unless the result is an exact copy of the learnt model (score is 3).

3. Both categorical and numerical data
   The overall interpretability score, interpretability scores for the visualisation application purpose and recommendation score depend on the ratio of categorical to numerical features.

#### 5.3.2.6 Time-series data

Time-series data can consist of categorical data (0s and 1s), numerical data or a mixture of both. The overall interpretability score, the interpretability score for the visualisation application purpose and recommendation score are similar to those of the transactional data.

## 5.4 The developed decision support framework

In this section the developed decision support framework is presented. Firstly, an example scenario is presented to illustrate how to interpret and use the developed decision support framework. Then the complete developed decision support framework is presented.

Table 5.5: A small example for audio data

| Application Type | NC | LogReg LIBLIN |
|---|---|---|
| association | 2 | 2 |
| classification | | |
| clustering | | |
| data compression | | 3 |
| outlier detection | 2 | 2 |
| visualisation | Y | Y |

### 5.4.1 An explanatory example

The following scenario is presented to illustrate how to interpret the developed decision support framework. For details on how the scores were determined, please refer to the previous sections.

In Table 5.5 a small snippet for audio data is presented. The application purposes are listed on the vertical axis and the ML algorithms on the horizontal axis at the top of the table.

The application purposes are **association**, **classification**, **clustering**, **data compression**, **outlier detection** and **visualisation**. The ML algorithms are NC and LogReg with LIBLIN solver.

Interpretation of the snippet works as follows:

To achieve **association** with NC, little information is given by the algorithm and programming would be required to achieve association, resulting in a yellow programming score. The results of the programming task does not model the algorithm perfectly and will be useful as a guideline only; therefore, the recommendation score is 2. To achieve **association** with LogReg with LIBLIN solver, the algorithm only provides the predicted labels and more programming would be required to achieve association, resulting in an orange programming score. The results of the programming task does not model the algorithm perfectly and will be useful as a guideline only; therefore, the recommendation score is 2.

Both algorithms perform **classification**; thus, the programming score is dark green for both. Neither of the algorithms can perform **clustering**; thus, the programming

score is white for both.

For **data compression** NC provides the necessary output and the score is dark green. Little programming would be required for LogReg with LIBLIN solver and the programming score is light green. The results of the programming task replicate the developed model and is reliable to use in future applications; thus, the recommendation score is 3.

NC can perform **outlier detection** by manually programming it using additional output provided by the developed model; thus, the programming score is yellow. LogReg with LIBLIN solver performs **outlier detection**; however, it requires programming using the labels outputted by the algorithm and the programming score is orange. In both cases the results of the programming task should only be used as a guideline; therefore, the recommendation score is 2.

**Visualisation** for both algorithms is possible by using the labels outputted by the algorithms and programming; therefore, the programming score is orange. The interpretability score for visualisation is Y, since correlations can be made from the data.

### 5.4.2 The developed decision support framework

The developed decision support framework is presented in this section using figures. Each figure illustrates a cut or section of the framework along the data type axis of the framework. To simplify the sections, they were further divided into the three different classes of ML algorithms employed in this research: clustering, classification and regression. Each section presents the application purposes and ML algorithms per data type and ML algorithm class.

A total of 14 different figures are presented (six for clustering, six for classification and two for regression). Figure 5.5 provides a guide for the framework to navigate to the correct framework section given the data type and ML algorithm class.

The analyst or user should follow the following steps to locate the appropriate ML algorithm given their data type and application purpose:

1. Identify the data type.

2. Identify the class of the ML algorithm, *i.e* whether it is clustering, classification or regression. The questions at the top of Figure 5.5 can assist the analyst.

3. Locate the applicable framework section by using Figure 5.5 and the previously identified information.
4. On the framework section, identify the desired application purpose of the application on the vertical axis.
5. Follow the application purpose horizontally to identify the best score for it.
6. Identify the appropriate ML algorithm by reading vertically upwards from the identified score until the name of the ML algorithm appears.

After locating the appropriate ML algorithm, it is the analyst's responsibility to investigate the chosen ML algorithm in depth, including the required data preprocessing and parameter tuning aspects to ensure that they develop a reliable model.

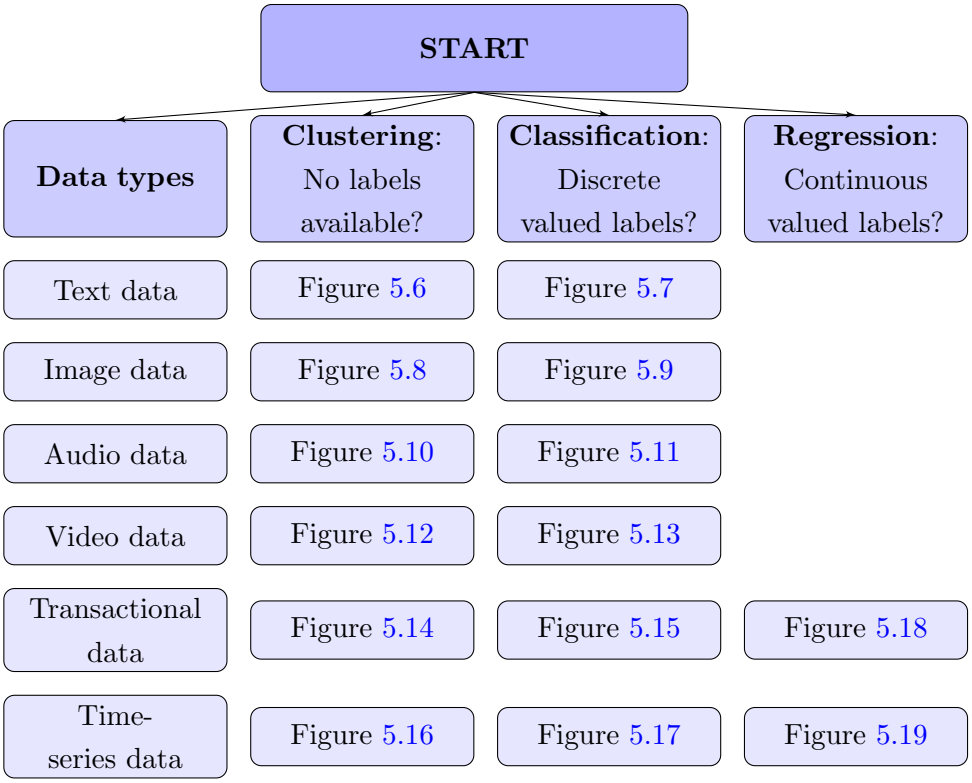


Figure 5.5: A guide for the developed decision support framework

| Interpretability of the data type: | N |
|---|---|

| | Application Type | AHC average | AHC complete | AHC single | AHC ward | DBSCAN ball-tree | DBSCAN brute | DBSCAN KD-tree | KMC | miniKMC | MS | SVM LIN | SVM POLY | SVM RBF | SVM SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | classification | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | |
| 4 | data compression | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | |
| 7 | outlier detection | 0 | 0 | 0 | 0 | | | | 0 | 0 | 0 | | | | |
| 8 | regression | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | | | | | | | | | | |
| | Performance (FMI) | 0.5431 | 0.5004 | 0.5449 | 0.5065 | 0.4745 | 0.4745 | 0.4745 | 0.4546 | 0.4570 | 0.5466 | 0.4015 | 0.4528 | 0.4041 | 0.4015 |
| | Performance rank | 3 | 5 | 2 | 4 | 7 | 7 | 7 | 10 | 9 | 1 | 14 | 11 | 13 | 14 |
| | Time (seconds) | 131.4 | 131.3 | 130.6 | 131.4 | 1.4 | 274.6 | 88.3 | 46.4 | 1.5 | 3820.1 | 372.7 | 260.5 | 326.0 | 372.7 |
| | Time rank | 7 | 6 | 5 | 8 | 1 | 10 | 4 | 3 | 2 | 14 | 13 | 9 | 12 | 13 |

Figure 5.6: The framework section for the clustering of text data

| Interpretability of the data type: | N |
|---|---|

| | Application Type | CART | ET | EF10 | EF100 | ID3 | RF10 | RF100 | KNN auto | KNN ball-tree | KNN brute | KNN KD-tree | NC | LogReg LFBGS | LogReg LIBLIN | LogReg NTCG | LogReg SAG | LogReg SAGA | MLP ADAM identity | MLP ADAM logistic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | classification | | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | regression | | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | N | N | N | N | N | N | N | | | | | | | | | | | | |
| | Performance (Acc.) | 0.6435 | 0.6018 | 0.6671 | 0.6968 | 0.6269 | 0.6612 | 0.6933 | 0.5767 | 0.5804 | 0.5766 | 0.5767 | 0.6412 | 0.6804 | 0.6855 | 0.6804 | 0.6803 | 0.6804 | 0.6752 | 0.6834 |
| | Performance rank | 22 | 29 | 16 | 2 | 26 | 18 | 3 | 31.5 | 30 | 33 | 31.5 | 24 | 7 | 4 | 7 | 9 | 7 | 11 | 5 |
| | Time (seconds) | 24.8 | 0.6 | 5.3 | 52.1 | 577.1 | 1.7 | 16.7 | 210.1 | 175.9 | 1.3 | 339.9 | 0.4 | 7.5 | 0.6 | 22.4 | 65.0 | 61.9 | 25.1 | 69.9 |
| | Time rank | 14 | 4 | 9 | 22 | 37 | 8 | 12 | 30 | 29 | 7 | 35 | 1 | 10 | 5 | 13 | 24 | 23 | 15 | 25 |

| | Application Type | MLP ADAM ReLu | MLP ADAM tanh | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | BNB | CNB | GNB | MNB | SVC LIN | SVC POLY | SVC RBF | SVC SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | classification | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | regression | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | | | | | | | | | | | | | | | | | | |
| | Performance (Acc.) | 0.6635 | 0.6772 | 0.6468 | 0.6718 | 0.6420 | 0.6589 | 0.6689 | 0.6131 | 0.6696 | 0.6696 | 0.6321 | 0.6140 | 0.5371 | 0.6462 | 0.6992 | 0.4430 | 0.4430 | 0.4430 | 0.4430 |
| | Performance rank | 17 | 10 | 20 | 12 | 23 | 19 | 15 | 28 | 13 | 14 | 25 | 27 | 34 | 21 | 1 | 36 | 36 | 36 | |
| | Time (seconds) | 27.9 | 32.7 | 41.4 | 46.5 | 48.8 | 42.4 | 171.8 | 103.2 | 251.5 | 173.2 | 0.7 | 0.4 | 7.8 | 0.4 | 318.9 | 254.6 | 256.5 | 463.5 | |
| | Time rank | 16 | 17 | 18 | 20 | 21 | 19 | 27 | 26 | 31 | 28 | 6 | 2 | 11 | 3 | 34 | 32 | 33 | 36 | |

Figure 5.7: The framework section for the classification of text data

| Interpretability of the data type: | N |
|---|---|

| | Application Type | AHC average | AHC complete | AHC single | AHC ward | DBSCAN ball-tree | DBSCAN brute | DBSCAN KD-tree | KMC | miniKMC | MS | SVM LIN | SVM POLY | SVM RBF | SVM SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | classification | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | |
| 4 | data compression | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | |
| 7 | outlier detection | 0 | 0 | 0 | 0 | | | | 0 | 0 | 0 | | | | |
| 8 | regression | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N |
| | Performance (FMI) | 0.4592 | 0.3350 | 0.4434 | 0.3775 | 0.441 | 0.441 | 0.441 | 0.3513 | 0.3418 | 0.4467 | 0.3293 | 0.3290 | 0.3255 | 0.3297 |
| | Performance rank | 1 | 10 | 3 | 7 | 5 | 5 | 5 | 8 | 9 | 2 | 12 | 13 | 14 | 11 |
| | Time (seconds) | 112.2 | 111.4 | 104.3 | 112.3 | 5.1 | 165.6 | 181.8 | 16.7 | 1.0 | 11651.2 | 321.1 | 321.9 | 336.4 | 323.6 |
| | Time rank | 6 | 5 | 4 | 7 | 2 | 8 | 9 | 3 | 1 | 14 | 10 | 11 | 13 | 12 |

Figure 5.8: The framework section for the clustering of image data

| Interpretability of the data type: | N |
|---|---|

| Application Type | CART | ET | EF10 | EF100 | ID3 | RF10 | RF100 | KNN auto | KNN ball-tree | KNN brute | KNN KD-tree | NC | LogReg LFBGS | LogReg LIBLIN | LogReg NTCG | LogReg SAG | LogReg SAGA | MLP ADAM identity | MLP ADAM logistic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 classification | | | | | | | | | | | | | | | | | | | |
| 3 clustering | | | | | | | | | | | | | | | | | | | |
| 4 data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 optimisation | | | | | | | | | | | | | | | | | | | |
| 7 outlier detection | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 regression | | | | | | | | | | | | | | | | | | | |
| 9 reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 simulations | | | | | | | | | | | | | | | | | | | |
| 11 summation | | | | | | | | | | | | | | | | | | | |
| 12 transparency | | | | | | | | | | | | | | | | | | | |
| 13 visualisation | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Performance (Acc.) | 0.4950 | 0.4681 | 0.5915 | 0.6341 | 0.4842 | 0.5903 | 0.6367 | 0.6049 | 0.6049 | 0.6049 | 0.6049 | 0.5145 | 0.6247 | 0.6245 | 0.6245 | 0.6267 | 0.6267 | 0.6042 | 0.4657 |
| Performance rank | 29 | 32 | 17 | 2 | 31 | 18 | 1 | 10.5 | 10.5 | 10.5 | 10.5 | 25 | 6 | 8 | 7 | 4 | 3 | 13 | 33 |
| Time (seconds) | 3.5 | 0.1 | 0.7 | 6.0 | 667.3 | 1.2 | 11.7 | 537.1 | 489.0 | 16.6 | 537.1 | 0.2 | 10.4 | 29.0 | 32.0 | 90.4 | 107.6 | 7.8 | 10.7 |
| Time rank | 9 | 1 | 6 | 10 | 37 | 7 | 16 | 34 | 33 | 17 | 35 | 4 | 14 | 19 | 27 | 28 | 29 | 11 | 15 |

| Application Type | MLP ADAM ReLu | MLP ADAM tanh | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | BNB | CNB | GNB | MNB | SVC LIN | SVC POLY | SVC RBF | SVC SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 classification | | | | | | | | | | | | | | | | | | |
| 3 clustering | | | | | | | | | | | | | | | | | | |
| 4 data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 educational assist. | | | | | | | | | | | | | | | | | | |
| 6 optimisation | | | | | | | | | | | | | | | | | | |
| 7 outlier detection | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 regression | | | | | | | | | | | | | | | | | | |
| 9 reverse prediction | | | | | | | | | | | | | | | | | | |
| 10 simulations | | | | | | | | | | | | | | | | | | |
| 11 summation | | | | | | | | | | | | | | | | | | |
| 12 transparency | | | | | | | | | | | | | | | | | | |
| 13 visualisation | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| Performance (Acc.) | 0.3668 | 0.4496 | 0.6002 | 0.5149 | 0.5242 | 0.5529 | 0.5633 | 0.5926 | 0.5192 | 0.5965 | 0.5061 | 0.4047 | 0.5101 | 0.5531 | 0.6261 | 0.4648 | 0.5069 | 0.4599 |
| Performance rank | 37 | 35 | 14 | 24 | 22 | 21 | 19 | 16 | 23 | 15 | 28 | 36 | 26 | 20 | 5 | 34 | 27 | 30 |
| Time (seconds) | 8.8 | 8.2 | 36.0 | 45.4 | 39.9 | 44.6 | 18.7 | 48.6 | 29.5 | 37.0 | 0.1 | 0.3 | 2.2 | 0.1 | 185.2 | 661.5 | 351.2 | 406.3 |
| Time rank | 13 | 12 | 21 | 25 | 23 | 24 | 18 | 26 | 20 | 22 | 2 | 5 | 8 | 3 | 30 | 36 | 31 | 32 |

Figure 5.9: The framework section for the classification of image data

| Interpretability of the data type: | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Y | | | | | | | | | | | | | | | |

| | Application Type | AHC average | AHC complete | AHC single | AHC ward | DBSCAN ball-tree | DBSCAN brute | DBSCAN KD-tree | KMC | miniKMC | MS | SVM LIN | SVM POLY | SVM RBF | SVM SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | classification | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | |
| 4 | data compression | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | | | | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | |
| 7 | outlier detection | 2 | 2 | 2 | 2 | | | | 2 | 2 | 2 | | | | |
| 8 | regression | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Performance (FMI) | 0.5537 | 0.5752 | 0.5147 | 0.5807 | 0.0215 | 0.0215 | 0.0215 | 0.6268 | 0.6051 | 0.5015 | 0.5364 | 0.5364 | 0.4402 | 0.4693 |
| | Performance rank | 5 | 4 | 8 | 3 | 13 | 13 | 13 | 1 | 2 | 9 | 7 | 7 | 11 | 10 |
| | Time (seconds) | 45.9 | 46.5 | 23.0 | 55.8 | 17.4 | 1.5 | 2.2 | 1.0 | 0.2 | 428.9 | 40.2 | 44.6 | 84.9 | 61.7 |
| | Time rank | 9 | 10 | 6 | 11 | 5 | 3 | 4 | 2 | 1 | 14 | 7 | 8 | 13 | 12 |

Figure 5.10: The framework section for the clustering of audio data

| Interpretability of the data type: | Y |
|---|---|

| | Application Type | CART | ET | EF10 | EF100 | ID3 | RF10 | RF100 | KNN auto | KNN ball-tree | KNN brute | KNN KD-tree | KNN NC | LogReg LFBGS | LogReg LIBLIN | LogReg NTCG | LogReg SAG | LogReg SAGA | MLP ADAM identity | MLP ADAM logistic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | classification | | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | regression | | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Performance (Acc.) | 0.6996 | 0.4091 | 0.4613 | 0.6372 | 0.4772 | 0.5867 | 0.6721 | 0.7463 | 0.7463 | 0.7463 | 0.7463 | 0.6496 | 0.5032 | 0.4836 | 0.5243 | 0.5133 | 0.5300 | 0.6460 | 0.6182 |
| | Performance rank | 5 | 28 | 25 | 10 | 24 | 13 | 6 | 2.5 | 2.5 | 2.5 | 2.5 | 7 | 22 | 23 | 19 | 21 | 18 | 8 | 11 |
| | Time (seconds) | 0.79 | 0.08 | 0.72 | 7.01 | 574.22 | 2.04 | 20.31 | 2.44 | 1.62 | 115.05 | 2.43 | 0.04 | 3.84 | 3.69 | 18.36 | 50.57 | 52.83 | 9.50 | 14.23 |
| | Time rank | 8 | 5 | 7 | 16 | 33 | 10 | 23 | 12 | 9 | 32 | 11 | 3 | 14 | 13 | 22 | 30 | 31 | 19 | 21 |

| | Application Type | MLP ADAM ReLu | MLP ADAM tanh | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | BNB | CNB | GNB | MNB | SVC LIN | SVC POLY | SVC RBF | SVC SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | | | |
| 2 | classification | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | | | | |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | | | |
| 8 | regression | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | | | | |
| | Performance (Acc.) | 0.6424 | 0.6092 | 0.5199 | 0.5527 | 0.5459 | 0.5541 | 0.3394 | 0.3972 | 0.3298 | 0.3684 | 0.2752 | 0.4412 | 0.4363 | 0.5367 | | | | |
| | Performance rank | 9 | 12 | 20 | 15 | 16 | 14 | 31 | 29 | 32 | 30 | 33 | 27 | 26 | 17 | | | | |
| | Time (seconds) | 23.08 | 10.90 | 26.96 | 48.54 | 29.00 | 34.62 | 34.87 | 8.31 | 7.61 | 4.95 | 0.07 | 0.04 | 0.31 | 0.04 | | | | |
| | Time rank | 24 | 20 | 25 | 29 | 26 | 27 | 28 | 18 | 17 | 15 | 4 | 1 | 6 | 2 | | | | |

Figure 5.11: The framework section for the classification of audio data

| Interpretability of the data type: | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | | | | | | | | | | | | | | | | | |
| | Application | AHC | AHC | AHC | AHC | DBSCAN | DBSCAN | DBSCAN | KMC | miniKMC | MS | SVM | SVM | SVM | SVM | | | |
| | Type | average | complete | single | ward | ball-tree | brute | KD-tree | | | | LIN | POLY | RBF | SIG | | | |
| 1 | association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 2 | classification | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | |
| 4 | data compression | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | 3 | 3 | 3 | 3 | | | |
| 5 | educational assist. | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 0 | 0 | 0 | 0 | | | | 0 | 0 | 0 | | | | | | | |
| 8 | regression | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | | | |
| | Performance (FMI) | 0.5694 | 0.5537 | 0.5382 | 0.5653 | 0.5422 | 0.5422 | 0.5422 | 0.5221 | 0.5304 | 0.5319 | 0.4790 | 0.4519 | 0.4120 | 0.4933 | | | |
| | Performance rank | 1 | 3 | 7 | 2 | 5 | 5 | 5 | 10 | 9 | 8 | 12 | 13 | 14 | 11 | | | |
| | Time (seconds) | 33.5 | 33.5 | 33.5 | 33.6 | 0.7 | 20.2 | 11.6 | 23.0 | 4.8 | 6352.0 | 86.8 | 86.8 | 84.3 | 86.6 | | | |
| | Time rank | 7 | 6 | 8 | 9 | 1 | 4 | 3 | 5 | 2 | 14 | 13 | 12 | 10 | 11 | | | |

Figure 5.12: The framework section for the clustering of video data

Figure 5.13: The framework section for the classification of video data

| Interpretability of the data type: | Y |
|---|---|

| Categorical / Numerical values |
|---|

| Application Type | AHC average | AHC complete | AHC single | AHC ward | DBSCAN ball-tree | DBSCAN brute | DBSCAN KD-tree | KMC | miniKMC | MS | SVM LIN | SVM POLY | SVM RBF | SVM SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 association | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 classification | | | | | | | | | | | | | | |
| 3 clustering | | | | | | | | | | | | | | |
| 4 data compression | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | | | | 3 | 3 | 3 | 3 |
| 5 educational assist. | | | | | | | | | | | | | | |
| 6 optimisation | | | | | | | | | | | | | | |
| 7 outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | | | | 0/2 | 0/2 | 0/2 | | | | |
| 8 regression | | | | | | | | | | | | | | |
| 9 reverse prediction | | | | | | | | | | | | | | |
| 10 simulations | | | | | | | | | | | | | | |
| 11 summation | | | | | | | | | | | | | | |
| 12 transparency | | | | | | | | | | | | | | |
| 13 visualisation | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| Performance (FMI) | 0.6909 | 0.6430 | 0.7035 | 0.5977 | 0.5678 | 0.5678 | 0.5678 | 0.5848 | 0.5745 | 0.6977 | 0.5622 | 0.56780 | 0.5196 | 0.5450 |
| Performance rank | 3 | 4 | 1 | 5 | 10 | 10 | 10 | 6 | 7 | 2 | 12 | 8 | 14 | 13 |
| Time (seconds) | 22.50 | 22.39 | 13.99 | 22.61 | 3.71 | 31.65 | 23.60 | 0.36 | 0.05 | 975.09 | 37.12 | 39.07 | 50.08 | 43.77 |
| Time rank | 6 | 5 | 4 | 7 | 3 | 9 | 8 | 2 | 1 | 14 | 10 | 11 | 13 | 12 |

Figure 5.14: The framework section for the clustering of transactional data

Interpretability of the data type: Y

Categorical / Numerical values

| Application Type | CART | ET | EF10 | EF100 | ID3 | RF10 | RF100 | KNN auto | KNN ball-tree | KNN brute | KNN KD-tree | NC | LogReg LFBGS | LogReg LIBLIN | LogReg NTCG | LogReg SAG | LogReg SAGA | MLP ADAM identity | MLP ADAM logistic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 classification | | | | | | | | | | | | | | | | | | | |
| 3 clustering | | | | | | | | | | | | | | | | | | | |
| 4 data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 optimisation | | | | | | | | | | | | | | | | | | | |
| 7 outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 8 regression | | | | | | | | | | | | | | | | | | | |
| 9 reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 simulations | | | | | | | | | | | | | | | | | | | |
| 11 summation | | | | | | | | | | | | | | | | | | | |
| 12 transparency | | | | | | | | | | | | | | | | | | | |
| 13 visualisation | Y | Y | Y | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| Performance (Acc.) | 0.8584 | 0.8307 | 0.8713 | 0.8529 | 0.8579 | 0.8693 | 0.8881 | 0.8432 | 0.8431 | 0.8400 | 0.8432 | 0.7631 | 0.8577 | 0.8535 | 0.8577 | 0.8577 | 0.8573 | 0.8698 | 0.8683 |
| Performance rank | 17 | 28 | 4 | 2 | 18 | 8 | 1 | 24.5 | 26 | 27 | 24.5 | 34 | 21 | 23 | 19.5 | 19.5 | 22 | 7 | 11 |
| Time (seconds) | 0.98 | 0.02 | 0.21 | 1.96 | 165.58 | 0.42 | 4.06 | 26.25 | 51.23 | 8.69 | 26.25 | 0.01 | 0.15 | 0.09 | 0.23 | 0.51 | 0.97 | 0.86 | 3.19 |
| Time rank | 16 | 4 | 9 | 19 | 37 | 11 | 27 | 34 | 36 | 30 | 35 | 3 | 8 | 7 | 10 | 13 | 15 | 14 | 25 |

| Application Type | MLP ADAM ReLu | MLP ADAM tanh | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | BNB | CNB | GNB | MNB | SVC LIN | SVC POLY | SVC RBF | SVC SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 association | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 classification | | | | | | | | | | | | | | | | | | |
| 3 clustering | | | | | | | | | | | | | | | | | | |
| 4 data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 5 educational assist. | | | | | | | | | | | | | | | | | | |
| 6 optimisation | | | | | | | | | | | | | | | | | | |
| 7 outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 8 regression | | | | | | | | | | | | | | | | | | |
| 9 reverse prediction | | | | | | | | | | | | | | | | | | |
| 10 simulations | | | | | | | | | | | | | | | | | | |
| 11 summation | | | | | | | | | | | | | | | | | | |
| 12 transparency | | | | | | | | | | | | | | | | | | |
| 13 visualisation | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| Performance (Acc.) | 0.8691 | 0.8700 | 0.8612 | 0.8596 | 0.8668 | 0.8623 | 0.8699 | 0.8060 | 0.8728 | 0.8689 | 0.7808 | 0.7471 | 0.7289 | 0.7995 | 0.8630 | 0.7113 | 0.8083 | 0.7904 |
| Performance rank | 9 | 5 | 15 | 16 | 12 | 14 | 6 | 30 | 3 | 10 | 33 | 35 | 36 | 31 | 13 | 37 | 29 | 32 |
| Time (seconds) | 2.56 | 2.79 | 0.46 | 1.56 | 2.11 | 2.44 | 1.63 | 4.20 | 2.79 | 3.88 | 0.03 | 0.01 | 0.05 | 0.01 | 6.94 | 8.74 | 9.21 | 9.16 |
| Time rank | 22 | 24 | 12 | 17 | 20 | 21 | 18 | 28 | 23 | 26 | 5 | 2 | 6 | 1 | 29 | 31 | 33 | 32 |

Figure 5.15: The framework section for the classification of transactional data

| Interpretability of the data type: | Y |
|---|---|

| Categorical / Numerical values |
|---|

| | Application | AHC | AHC | AHC | AHC | DBSCAN | DBSCAN | DBSCAN | KMC | miniKMC | MS | SVM | SVM | SVM | SVM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Type | average | complete | single | ward | ball-tree | brute | KD-tree | | | | LIN | POLY | RBF | SIG |
| 1 | association | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 | classification | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | |
| 4 | data compression | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | | | | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | |
| 7 | outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | | | | 0/2 | 0/2 | 0/2 | | | | |
| 8 | regression | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (FMI) | 0.4816 | 0.4382 | 0.5720 | 0.5093 | 0.4699 | 0.4699 | 0.4699 | 0.4960 | 0.4951 | 0.5080 | 0.5191 | 0.5213 | 0.4979 | 0.5640 |
| | Performance rank | 10 | 14 | 1 | 5 | 12 | 12 | 12 | 8 | 9 | 6 | 4 | 3 | 7 | 2 |
| | Time (seconds) | 19.28 | 19.28 | 13.60 | 21.75 | 6.32 | 0.77 | 2.12 | 0.98 | 0.09 | 845.88 | 22.55 | 24.77 | 102.66 | 24.24 |
| | Time rank | 7 | 8 | 6 | 9 | 5 | 2 | 4 | 3 | 1 | 14 | 10 | 12 | 13 | 11 |

Figure 5.16: The framework section for the clustering of time-series data

Interpretability of the data type: Y

Categorical / Numerical values

| | Application / Type | CART | ET | EF10 | EF100 | ID3 | RF10 | RF100 | KNN auto | KNN ball-tree | KNN brute | KNN KD-tree | NC | LogReg LFBGS | LogReg LIBLIN | LogReg NTCG | LogReg SAG | LogReg SAGA | MLP ADAM identity | MLP ADAM logistic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 | classification | | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 8 | regression | | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (Acc.) | 0.8342 | 0.7291 | 0.7994 | 0.8608 | 0.7370 | 0.7859 | 0.8264 | 0.6727 | 0.6722 | 0.6723 | 0.6727 | 0.4568 | 0.8305 | 0.8347 | 0.8314 | 0.8039 | 0.8034 | 0.8248 | 0.8264 |
| | Performance rank | 4 | 22 | 14 | 1 | 20 | 17 | 7 | 26.5 | 29 | 28 | 26.5 | 37 | 6 | 3 | 5 | 11 | 12 | 9 | 8 |
| | Time (seconds) | 0.48 | 0.02 | 0.16 | 1.45 | 334.63 | 0.57 | 5.66 | 1.13 | 1.10 | 43.26 | 1.15 | 0.10 | 0.67 | 1.04 | 2.59+ | 5.39 | 6.16 | 1.56 | 1.51 |
| | Time rank | 8 | 3 | 7 | 18 | 36 | 9 | 29 | 15 | 13 | 32 | 14 | 5 | 10 | 12 | 24 | 27 | 30 | 21 | 20 |

| | Application / Type | MLP ADAM ReLu | MLP ADAM tanh | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | BNB | CNB | GNB | MNB | SVC LIN | SVC POLY | SVC RBF | SVC SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 2 | classification | | | | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 5 | educational assist. | | | | | | | | | | | | | | | | | | | |
| 6 | optimisation | | | | | | | | | | | | | | | | | | | |
| 7 | outlier detection | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| 8 | regression | | | | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | | | | | | | | | | | | | | | | | | | |
| 10 | simulations | | | | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | | | | |
| 13 | visualisation | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (Acc.) | 0.8185 | 0.8005 | 0.7884 | 0.7291 | 0.7884 | 0.7592 | 0.6191 | 0.5479 | 0.5650 | 0.5626 | 0.7267 | 0.6970 | 0.7675 | 0.7141 | 0.8364 | 0.6211 | 0.6062 | 0.4593 |
| | Performance rank | 10 | 13 | 15.5 | 21 | 15.5 | 19 | 31 | 35 | 33 | 34 | 23 | 25 | 18 | 24 | 2 | 30 | 32 | 36 | |
| | Time (seconds) | 1.85 | 1.46 | 1.83 | 5.52 | 2.88 | 4.64 | 10.09 | 1.18 | 0.93 | 1.38 | 0.05 | 0.01 | 0.12 | 0.01 | 127.59 | 0265.46 | 245.68 | 238.12 | |
| | Time rank | 23 | 19 | 22 | 28 | 25 | 26 | 31 | 16 | 11 | 17 | 4 | 2 | 6 | 1 | 33 | 37 | 35 | 34 | |

Figure 5.17: The framework section for the classification of time-series data

Interpretability of the data type: Y

Categorical / Numerical values

| | Application Type | CART | ET | EF 10 | EF 100 | ID3 | RF 10 | RF 100 | KNR auto | KNR ball-tree | KNR brute | KNR KD-tree | LinReg | MLP ADAM identity | MLP ADAM logistic | MLP ADAM ReLu | MLP ADAM tanh |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | | | | | | | | |
| 2 | classification | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | | | | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | |
| 6 | optimisation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | regression | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 10 | simulations | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (MSE) | 0.0151 | 0.0151 | 0.0099 | 0.0095 | 0.0274 | 0.0088 | 0.008 | 0.0158 | 0.0161 | 0.0157 | 0.0158 | 3.9e+22 | 0.0197 | 0.0207 | 0.0143 | 0.0167 |
| | Performance rank | 8 | 9 | 4 | 3 | 25 | 2 | 1 | 12.5 | 14 | 11 | 12.5 | 32 | 21 | 22 | 6 | 15 |
| | Time (seconds) | 0.029 | 0.008 | 0.083 | 0.818 | 5.556 | 0.153 | 1.814 | 0.097 | 0.166 | 0.858 | 0.097 | 0.005 | 0.162 | 0.369 | 0.238 | 0.244 |
| | Time rank | 3 | 2 | 4 | 23 | 32 | 13 | 31 | 6 | 11 | 24 | 5 | 1 | 10 | 18 | 15 | 16 |

| | Application Type | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | NUSVR LIN | NUSVR POLY | NUSVR RBF | NUSVR SIG | SVR LIN | SVR POLY | SVR RBF | SVR SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | | | | | | | | | | | | | | | | |
| 2 | classification | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | |
| 6 | optimisation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | regression | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 10 | simulations | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | |
| 13 | visualisation | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (MSE) | 0.0169 | 0.0137 | 0.0149 | 0.0157 | 0.0253 | 0.0423 | 0.0319 | 0.0261 | 0.0195 | 0.0382 | 0.0187 | 27.4987 | 0.0187 | 0.0371 | 0.0192 | 35.0581 |
| | Performance rank | 16 | 5 | 7 | 10 | 23 | 29 | 26 | 24 | 20 | 28 | 17 | 30 | 18 | 27 | 19 | 31 |
| | Time (seconds) | 0.146 | 0.323 | 0.446 | 0.593 | 0.156 | 0.123 | 0.187 | 0.176 | 1.131 | 0.893 | 1.321 | 1.439 | 0.521 | 0.671 | 0.871 | 1.428 |
| | Time rank | 8 | 17 | 19 | 21 | 9 | 7 | 14 | 12 | 27 | 26 | 28 | 30 | 20 | 22 | 25 | 29 |

Figure 5.18: The framework section for the regression of transactional data

| Interpretability of the data type: | Y |
|---|---|

Categorical / Numerical values

| | Application Type | CART | ET | EF`10 | EF`100 | ID3 | RF`10 | RF`100 | KNR auto | KNR ball-tree | KNR brute | KNR KD-tree | LinReg | MLP ADAM identity | MLP ADAM logistic | MLP ADAM ReLu | MLP ADAM tanh |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | | | | | | | | |
| 2 | classification | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | | | | | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | |
| 6 | optimisation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | regression | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 10 | simulations | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | |
| 13 | visualisation | Y | Y | Y | Y | Y | Y | Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (MSE) | 0.0263 | 0.0126 | 0.0081 | 0.0072 | 0.0325 | 0.0103 | 0.0095 | 0.0497 | 0.0497 | 0.0497 | 0.0497 | 0.0050 | 0.0077 | 0.0076 | 0.0092 | 0.0063 |
| | Performance rank | 23 | 19 | 11 | 7 | 24 | 16 | 14 | 26,5 | 26,5 | 26,5 | 26,5 | 2 | 9 | 8 | 13 | 6 |
| | Time (seconds) | 0.19 | 0.04 | 0.34 | 3.35 | 71.59 | 1.19 | 11.86 | 2.36 | 6.26 | 26.79 | 2.35 | 0.02 | 0.33 | 0.39 | 0.37 | 0.39 |
| | Time rank | 3 | 2 | 7 | 23 | 28 | 19 | 26 | 22 | 24 | 27 | 21 | 1 | 6 | 13 | 8 | 12 |

| | Application Type | MLP LFBGS identity | MLP LFBGS logistic | MLP LFBGS ReLu | MLP LFBGS tanh | MLP SGD identity | MLP SGD Logistic | MLP SGD ReLu | MLP SGD tanh | NUSVR LIN | NUSVR POLY | NUSVR RBF | NUSVR SIG | SVR LIN | SVR POLY | SVR RBF | SVR SIG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | association | | | | | | | | | | | | | | | | |
| 2 | classification | | | | | | | | | | | | | | | | |
| 3 | clustering | | | | | | | | | | | | | | | | |
| 4 | data compression | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | educational assist. | | | | | | | | | | | | | | | | |
| 6 | optimisation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | outlier detection | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | regression | | | | | | | | | | | | | | | | |
| 9 | reverse prediction | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 10 | simulations | | | | | | | | | | | | | | | | |
| 11 | summation | | | | | | | | | | | | | | | | |
| 12 | transparency | | | | | | | | | | | | | | | | |
| 13 | visualisation | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y | N/Y |
| | Performance (MSE) | 0.0051 | 0.0051 | 0.1253 | 0.5163 | 0.0080 | 0.0170 | 0.0123 | 0.0095 | 0.0050 | 0.0192 | 0.0054 | 9.3478 | 0.0083 | 0.0132 | 0.0103 | 2.1426 |
| | Performance rank | 3 | 4 | 29 | 30 | 10 | 21 | 18 | 15 | 1 | 22 | 5 | 32 | 12 | 20 | 17 | 31 |
| | Time (seconds) | 0.31 | 0.39 | 0.49 | 0.63 | 0.31 | 0.49 | 0.39 | 0.39 | 148.54 | 172.13 | 226.02 | 215.46 | 0.56 | 1.87 | 0.81 | 9.70 |
| | Time rank | 5 | 11 | 14 | 17 | 4 | 15 | 10 | 9 | 29 | 30 | 32 | 31 | 16 | 20 | 18 | 25 |

Figure 5.19: The framework section for the regression of times-series data

## 5.5 Conclusion: Chapter 5

This chapter presented and discussed the process of developing the decision support framework for this research study. It introduced the conceptual framework and the five criteria of the framework. It presented the process of developing and deploying the machine learning algorithms used to further the framework. The investigation of the developed models to populate the framework were presented as well. Lastly, it presented the developed decision support framework.

The following chapter will focus on the validation of the developed decision support framework, by providing the feedback of the subject matter experts and possible end-users.

# Chapter 6

# Validation of the developed framework

In this chapter Phase 7 of Jabareen's methodology, 'validate the framework', will be applied. The developed decision support framework is validated by consulting subject matter experts (SMEs) and possible end-users (PEUs). In this chapter their feedback is interpreted and synthesised.

## 6.1 The subject matter experts for this study

A total of 6 SMEs were consulted to evaluate the technical and quantitative aspects of the developed framework. These SMEs were selected based in their experience and expertise. They were provided with a document presenting the goal of the study, a summary of the machine learning (ML) algorithms, the different data types, the definitions of the application purposes, the five criteria of the framework, the assumptions and limitations of the study and the developed decision support framework. The investigation of each ML algorithm was presented along with the reasoning used to allocate the five criteria to the ML algorithms, data types and application purposes.

The SMEs were asked to provide their opinions on the following aspects: the definitions of the application purposes, dataset preprocessing, ML algorithm implementations and the five criteria of the framework. Next, they were asked their opinions regarding the evaluation and allocation of the five criteria of the framework.

The following people were SMEs for this research study:

1. J. Du Preez, Professor in speech processing and pattern recognition, Department of Electrical and Electronic Engineering, Stellenbosch University

2. J. Grobler, Associate professor, Department of Industrial Engineering, Stellenbosch University

   Prof Grobler has published more than 15 journal articles and 15 conference papers in the research areas of data management, data mining (DM) and optimisation heuristics (Google scholar profile).

3. M. Kidd, Professor, Director: Centre of Statistical Consultation (CSC), Stellenbosch University (CSC).

   Prof Kidd has published over 100 journal articles and 27 conference papers in the research areas of statistics, DM, optimisation heuristics and ML.

4. A. Pretorius, PhD student, Department of Computer Science, Stellenbosch University

   Dr Pretorius has completed his PhD titled 'On noise regularised neural networks: initialisation, learning and inference'. He has published 12 articles in the research areas of artificial intelligence (AI), ML and DM (Google scholar profile).

5. M. Hoffmann, CEO and co-founder of Praelexis, Technopark, Stellenbosch (Prealexis)

   Prealexis is an AI and ML company which provides ML consultation and solutions in the banking, eduction, insurgence and healthcare sectors. The company was founded in 2013. Dr Hoffmann has published 4 journals and 11 conference papers in the research areas of ML and computer vision.

6. B. Herbst, Employee at Prealexis, former professor at Stellenbosch University (Prealexis).

   Prof Herbst has published more than 60 journal articles and 50 conference papers in the research areas of applied mathematics, AI, ML, computer vision and pattern recognition.

In the following sections their opinions, acknowledgements and criticisms are summarised and addressed.

### 6.1.1 The application purposes

The SMEs were questioned on the applicability and definitions of the application purposes. The following criticisms were provided:

1. Some of the application purposes are somewhat related, for example, association and clustering (Du Preez, 2019).
2. The definition of detection in this work is different to ML practice. Detection is to determine whether something is present or not in the dataset. The definition used in the study is a subset of this definition (Du Preez, 2019; Herbst, 2019).
3. The definition of optimisation in this study differs from ML practice. In ML practice, optimisation is the process of changing the parameters over time by using an algorithm or solver (Herbst, 2019; Pretorius, 2019).
4. The definition of simulation is slightly different than in practice. In this work it is implied as casual modelling (Hoffmann, 2019).

The criticisms were addressed by stating that the definitions of the application purposes are in context of this research study. For example, to a data scientist optimisation has to do with the learning of a model to increase its performance. In this study the definition of optimisation is in the context of industrial engineering, namely finding the optimal solution to a problem or question. Furthermore, the name of the application purpose of detection was adjusted to 'outlier detection' since it better describes the detection used in the context of this research study.

### 6.1.2 The dataset preprocessing

The SMEs were questioned on the applicability of the preprocessing performed on the datasets. The following acknowledgements were provided:

1. Good missing value handling, especially for time-series data (Grobler, 2019).
2. Good standardisation (Grobler, 2019).
3. The variety of datasets is a good choice to evaluate the ML algorithm performance (Grobler, 2019).

The following criticisms were provided:

1. It is recommended to use balanced datasets to obtain unbiased results (Grobler, 2019; Pretorius, 2019).

2. It is recommended to scale data using the $z$-score. The standardised method used in this study emphasises outliers (Du Preez, 2019).

3. It is unlike common practice to transform the target variables of regression to the $[0, 1]$ value range (Kidd, 2019; Pretorius, 2019). A user or analyst would like to make predictions on the scale that matters, *i.e.* the scale of the target variable. An additional transformation is needed to interpret the prediction. It might influence the underlying prediction structure of the developed models (Pretorius, 2019).

4. More advanced ML algorithms, including feature descriptions as a preprocessing step, have been designed specifically for image and video data, where data transformation to one-dimensional feature vectors is not necessary (Herbst, 2019; Hoffmann, 2019; Pretorius, 2019).

5. Dataset size has an influence on ML algorithm performance. An algorithm which is successful for a small dataset might not be as successful for a large dataset of the same data type (Pretorius, 2019).

6. Hoffmann (2019) and Herbst (2019) view the investigation of missing values as important to businesses.

In the future work section, suggestions were added to compare the performance of the ML algorithms on 1) both balanced and unbalanced datasets, 2) different input dataset formats and 3) different dataset sizes, to determine the significance of the difference. Another suggestion was added: to explore with more complex ML algorithms, including convolutional neural networks and deep learning.

The zero-to-one standardisation was used in this research study, since it allows the testing of the capabilities of the ML algorithms to perform outlier detection as an application purpose by emphasising outliers. As stated before, it also preserved the original data distributions as far as possible.

The target variables were only used to determine the performance of the ML algorithms; the values of the predictions were not further interpreted or used. Therefore, the scale of the target variable is not significant in this research study.

Missing value investigation was not part of the research study, since the goal of the study is not to evaluate a specific dataset(s) for the purpose of solving a specific business problem, as stated in Subsection 3.1.4.3. The goal of the study is to provide a general guideline of choosing an ML algorithm, regardless of the industry from which the dataset originates.

### 6.1.3 The machine learning algorithm implementations

The SMEs were questioned on the ML algorithm implementations. The following acknowledgements were provided:

1. It was a good choice to use *Python* since it provides a variety of useable code in one place (Du Preez, 2019; Hoffmann, 2019).

2. The performance of the ML algorithms was reliable, sound and comparable to those found in practice (Du Preez, 2019; Kidd, 2019).

3. The execution times of the ML algorithms were reliable, sound and comparable to those found in practice (Du Preez, 2019).

The following criticisms were provided:

1. Perhaps employ cross-validation when implementing repeated experiments (Grobler, 2019).

2. Parameter tuning has a significant effect on algorithm performance and is recommended, although it is noted that it is not part of the goal of the study (Grobler, 2019; Pretorius, 2019). Therefore, for this study the variability in the datasets determined the performance results of the ML algorithms (Pretorius, 2019). Hoffmann (2019) and Herbst (2019) strongly advised parameter tuning.

3. Due to the 'no-free-lunch' theorem, the performance of the algorithms with respect to accuracy can always be questioned. However, the additional criteria used to evaluate the algorithms partly addresses this theorem in the work (Grobler, 2019).

4. Only training and validation datasets were used; however, testing datasets are generally used to objectively evaluate the final performance of the developed models (Du Preez, 2019).

5. Investigate whether the employed packages run directly in *Python* or if it is implemented using another programming language since it has a significant influence on the execution times of the algorithms (Du Preez, 2019).

6. The programming score might be different if the ML algorithms were implemented in a different programming languages (Pretorius, 2019).

7. Every problem is unique. An algorithm which is successful for a data type might not be as successful for another dataset of the same data type (Hoffmann, 2019; Pretorius, 2019).

Prof Du Preez helped the researcher to better define the dataset division used in this study and the amendments were added to Subsection 5.2.2.8. In this research study validation datasets were not used since parameter tuning was not performed. Only training and testing datasets were used to train and objectively evaluate the performance of the developed models respectively.

Cross-validation was not employed since validation and parameter tuning was not performed in this research study. Only training and testing were performed. However, the use of cross-validation is a recommended practice and it was added to the future work section.

In the future work section a suggestion was made to compare the performance of the ML algorithms in different programming environments to determine the significance of the difference in terms of the five criteria of the framework.

The researcher is aware that parameter tuning is important during the development of ML algorithms; however, this was not part of the goal of the project. As stated, the developed framework is a guideline or is used to provide a starting point for the analyst. Thereafter, it is the analyst's responsibility to investigate the chosen ML algorithm in depth, including the required data preprocessing and parameter tuning aspects. The analyst's responsibilities were added to the small manual in Subsection 5.4.2.

Parameter tuning is not included in this research study, since the researcher wanted to determine which ML algorithms are robust and reliable, regardless of a specific

dataset. Parameter tuning focusses on the specific dataset to try to capture its underlying structure, for example learning the difference between Afrikaans and English text data or spam and not-spam email text data. In this study, datasets in general were approached, for example text or audio datasets.

### 6.1.4 The five criteria of the framework

The SMEs were questioned on the applicability, definitions and evaluation of the five criteria of the framework. The following acknowledgements were provided:

1. The explanatory example assisted the SMEs in understanding the interaction amongst the criteria and how to interpret the developed framework.
2. The criteria are applicable and well chosen (Grobler, 2019; Kidd, 2019). The criteria is applicable to semi-skilled analysts (Herbst, 2019; Hoffmann, 2019).
3. The interpretability and programming scores are sound, reliable and appropriately applied (Herbst, 2019; Hoffmann, 2019).
4. The use of the performance metrics for the clustering, classification and regression applications are appropriate for simplicity (Du Preez, 2019; Kidd, 2019; Pretorius, 2019).
5. Du Preez (2019), Grobler (2019)) and Kidd (2019) were satisfied with the different criteria and the reasoning on which the scores were allocated.

The following criticisms were provided:

1. Motivate why the Fowlkes-Mallows index (FMI) was used as the performance measure for clustering algorithms when there are many other clustering performance metrics available (Du Preez, 2019; Kidd, 2019).
2. Explain how the execution times and ranks were obtained (Grobler, 2019).
3. The execution times will differ depending on the computational power available to the analyst and if parallel processing is available. Some algorithms are more sequential than others. However, some algorithms can be easily parallelised, which make them ideal for situations where time is of essence (Pretorius, 2019).

4. Hoffmann (2019) and Herbst (2019) mentioned that in some cases training time is favoured, while in other cases scoring time is favoured. The execution time score in this work is the summation of both.

5. In the cases of clustering/detection problems, generally performance metrics such as recall, precision or F1-score is used (Grobler, 2019). The clustering metric depends on the application (Herbst, 2019; Hoffmann, 2019).

6. Other classification performance metrics are available, including cross-entropy, log-likelihood, precision and recall (Grobler, 2019; Herbst, 2019; Hoffmann, 2019).

7. The programming (Pretorius, 2019) and recommendation scores (Herbst, 2019; Hoffmann, 2019; Pretorius, 2019) were unclear.

The FMI was used, since after initial experimentation with the different clustering performance metrics FMI proved most comparable with the accuracy performance metric. The FMI requires the ground truth labels, which are available with clustering since the same datasets were used for both clustering and classification. It is also a simplistic measurement. As stated in Chapter 4, simplistic and popular classification and regression performance metrics were chosen for this study.

The explanation of the execution time and ranks was amended in Subsection 5.1.2 to improve the definition thereof. The execution times were provided as a basic indication of the time requirements of the ML algorithms, therefore training and scoring time were not separately indicated. The use of parallelisation is beyond the scope of this work as it requires more complex programming. For this study, the developed framework is designed to assist semi-skilled analysts with little experience in ML. The ML algorithms were implemented in a sequential fashion, one after the other, and this detail was added to Subsection 5.2.4.1.

The definitions of the programming and recommendation scores were amended in Subsection 5.1.2 to improve the understanding thereof.

### 6.1.5 General criticisms on the framework

The following general criticisms were provided:

1. Arrange the scores consistently such that higher values indicate better performers (Du Preez, 2019).

2. The framework is complex, perhaps try to simplify it in terms of the perspective of a practitioner (Du Preez, 2019; Kidd, 2019; Pretorius, 2019).

3. Hoffmann (2019) and Herbst (2019) mentioned that a analyst can achieve the same results by looking in a textbook or reading API (application programming interface) documentation.

In the developed framework the performance and execution time ranks were colour-coded to aid the analyst in quickly identifying the best, average and worst performers.

To aid a user of the framework, the researcher has added a small manual on how to use the developed framework in Subsection 5.4.2. A quick-link guide in the form of a figure, Figure 5.5, was added to quickly guide a user to the correct section of the developed framework. The responsibilities of the user were also included.

In conclusion, Hoffmann (2019) and Herbst (2019) disagreed on the performance of the ML algorithms due to lack of parameter tuning, however they do agree on the applicability of the framework as a good reference on the data type interpretability, programming aspects, the application purposes and ML algorithms.

In general, the SMEs found the framework acceptable and reliable. It provides a good reference on the data type interpretability, programming requirements, the application purposes, ML algorithms and performance trade-offs. The developed framework is condensed, summarised and enables a quick lookup of ML algorithms, compared to searching through textbooks or various API documentation, which might confuse the user. The framework indicates appropriate ML algorithms given the application purpose and data type of the user.

In the following section, the feedback of the PEUs are provided.

## 6.2 Possible end-users

A total of 12 possible end-users were consulted to determine whether the target audience can interpret and use the framework. The PEUs consisted of five final year B.Eng(Industrial), four M.Eng(Industrial) students, two working industrial engineers and a senior industrial engineering lecturer at Stellenbosch University. They all have knowledge and experience with programming. They were provided with a document presenting the goal of the study, a summary of the ML algorithms, the different data

types, the definitions of the application purposes, the five criteria of the framework, the assumptions and limitations of the study and lastly the developed decision support framework.

They were asked to provide their opinions on the following three aspects: the definitions of the application purposes, the five criteria of the framework and the use of framework. The questions asked whether these three aspects were clear, understandable, applicable, easy to interpret and easy to derive meaning from. They were also presented with three small exercises where they had to consult the framework to answer the related questions. Lastly, they were welcomed to provide any recommendations and criticisms.

The following people were PEUs for this research study:

1. Final year B.Eng(Industrial) students: J.M. Louw, S.H. Jordaan, A. Devenish, T. Jonker and J.R. Bosch
2. M.Eng(Industrial) students: A. Ellis, B. Leuvennink, K. Heyns and F. Adams
3. Industrial engineers: N. Kotzé and N. De Bruyn
4. Industrial engineering lecturer: T. Dirkse van Schalkwyk

All of the PEUs found the definitions of the application purposes and five criteria clear, understandable, applicable and useful. The following additional criteria were suggested: sample efficiency, the computational power required by each ML algorithm (big O notation) and whether the ML algorithms can be implemented or are accessible in a variety of programming languages, including R, Julia, C and C++. Due to time constraints, these criteria were not added to the developed decision support framework, but were added to the future research and recommendation section.

A PEU suggested to use either a ranking score or a performance value, not both, for the performance and execution time scores. The researcher decided to keep both and explained why both are available for the analyst. Some PEUs asked for better explanations of the recommendation score and the interpretability scores, especially the distinction between the interpretability for the data type as a whole and the interpretability score for the visualisation application purpose. The definitions were amended accordingly.

Two participating PEUs disagreed with the framework being understandable and clear. They mentioned that it is a lot of information to internalise and organise. Some PEUs found the framework congested and that it takes time to navigate through the framework. Suggestions included adding a small explanatory example, moving the legends of the scores closer to the framework and providing a navigation tool to help locate the appropriate framework section quicker. These suggestions were added to the previous chapter. Other suggestions included an *Excel* sheet or a graphical user interface (GUI), where the analyst enters the application purpose and data type, and the interface outputs the appropriate algorithm. This would save time and avoid mistakes. These suggestions were added to the future research section, since the goal of this work was to develop an initial conceptual decision support framework.

Other suggestions, which were implemented in the framework, were to provide a better distinction between the interpretability of the data type and the interpretability of the visualisation application purpose, provide a small legend on the framework to indicate what is meant by the '/' symbol in the recommendation score, and for consistency rather say 'performance' than using the FMI, accuracy and MSE scores.

The PEUs were presented with three different scenarios with questions, where they had to consult the framework to answer the questions. Scenario 1 one was based on the clustering of text data, Scenario 2 on the classification of video data and 3 on the regression of time-series data. The results of the answers are illustrated in Figure 6.1. The different colours indicate sub-questions, dark colours the correct answers and light colours the incorrect answers.

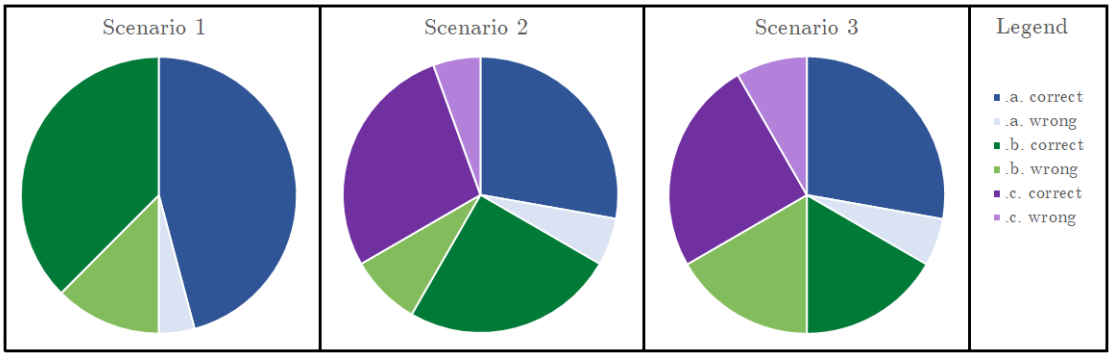


Figure 6.1: The results of the three scenarios

From the results it was clear that some PEUs did not understand the interpretability and programming scores. The definitions of these scores were altered to improve their understanding and explanatory examples were added to the documentation. With Scenario 3 sub-question 2, the question was misunderstood in terms of what it wanted the PEU to provide. This was due to a sentence structure error.

Overall, all of the PEUs mentioned that they would consult the framework when faced with an ML opportunity, whilst 11 mentioned that they would recommend it to a friend. One PEU wanted experience with the framework before recommending it to a friend. Some PEUs found the colours helpful as well. An explanatory example was added to the guideline before providing it to the final six PEUs, who found the example very helpful.

## 6.3 Conclusion: Chapter 6

This chapter presented the feedback of the subject matter experts and possible end-users. Their feedback was interpreted, synthesised and used to improve the developed decision support framework.

The following chapter will provide the final conclusions of this research study.

# Chapter 7

# Research summary and conclusions

In this chapter the research project is summarised and research conclusions are presented. Suggestions for future research are discussed and concluding remarks are presented.

## 7.1 Project summary and conclusion

**Chapter 1** detailed the problem background and motivation, problem statement, research objectives and research design. The research methodology, scope, limitations, assumptions and ethical considerations were discussed. The research assignment was presented in Section 1.3. The goal of this study was to develop and validate a *decision support framework* which considers both the *data characteristics* and the *application type* to enable small, medium and micro enterprises (SMMEs) to choose the appropriate machine learning (ML) algorithm for their unique data and application purpose. This study aimed to develop the framework for a semi-skilled analyst, with mathematics, statistics and programming education, who is familiar with the process of programming, yet has not specialised in the variety of ML algorithms which are available. The decision support framework provides proper guidance as to how to employ ML algorithms in a low-cost and useful way.

The project objectives were stated in Section 1.4 and included the provision of literature studies in data, data analytics (DA) and ML, as per **Objective 1**. The

developed decision support framework provides a variety of ML algorithms given the characteristics of the data and the application purpose, as per **Objective 2**. Furthermore, the framework is expanded to indicate the appropriate ML algorithm per data characteristic and application purpose pair, whilst considering the project management or iron triangle, as per **Objective 3**. Lastly, the developed framework indicates the relative trade-offs of the iron triangle per ML algorithm application, as per **Objective 4**.

In **Chapter 2** a literature study was conducted on the topic of frameworks. The literature study focused on three components of frameworks: the definition of a framework as per literature, the different types of frameworks and a methodology to develop a conceptual framework, namely Jabareen's framework development methodology. Given the literature study the definition of a framework for the purposes of this research study was developed and presented. Lastly, the presented framework development methodology was applied and aligned to the research methodology of this research study.

**Chapter 3** contained an in-depth literature review regarding DA. The types of DA; the application purposes of DA; the relationship between DA, data mining (DM) and ML; and two different processes of applying DA: the CRoss-Industry Standard Process for Data Mining (CRISP-DM) and the Sample, Explore, Modify, Model and Access (SEMMA) process, were explored and presented. The CRISP-DM was the process used throughout the study to perform ML. Based on the requirements to perform the CRISP-DM, a literature study on data and data preparation for ML applications was subsequently presented.

A literature review on ML was presented in **Chapter 4**. The definition of ML, types of ML, classes of ML and the variety of ML algorithms employed in this study were discussed. Applications in literature, algorithm methodology and the advantages and disadvantages of each algorithm were presented. Lastly, different performance metrics for each ML class were presented and the applicable metrics to this study were identified.

The literature studies of **Chapter 3** and **Chapter 4** were performed to satisfy **Objective 1** and they completed Phases 1 to 5 of Jabareen's framework development methodology.

**Chapter 5** discussed the creation and development of the decision support framework for this research study, as per Phase 6 of Jabareen's framework development

methodology. The conceptual framework was introduced, as well as the criteria used to assist the analyst in the final choice of which ML algorithm to choose. The framework was populated by applying Phase 3 and 4 of the CRISP-DM. The origin of the datasets utilised in this study, their preprocessing for the ML algorithms and the implementations of the ML algorithms were presented. The framework was furthered by evaluating different aspects of the developed ML algorithms in terms of the criteria of the framework. Lastly, the developed decision support framework was presented together with an explanatory example of how to use and interpret the decision support framework.

The *application purposes* provided in **Chapter 2**, the *machine learning algorithms* provided in **Chapter 4** and the *data characteristics or types* were used to develop the decision support framework and satisfy **Objectives 2, 3 and 4**.

**Chapter 6** reported on the validation of the developed framework, as per Phase 7 of Jabareen's framework development methodology. The framework was validated by consulting subject matter experts (SMEs) and possible end-users (PEUs).

## 7.2 Future research

DA and ML are broad fields which continuously develop and deepen as new discoveries are made and contributions are added to the fields. The research presented in this study utilises the capabilities of ML algorithms without considering the detail. The research can be further developed and refined by considering the following suggestions for further research:

- Experiment with more complex ML algorithms, including convolutional neural networks and deep learning.
- Experiment with the use of cross-validation to determine its influence on the performance of ML algorithms.
- Determine the relationship between dataset size and the computational cost or memory requirements per ML algorithm.
- Explore with different sized datasets, for example, 100 000 instances *vs* 1 million instances to determine the effect of different dataset sizes on the performance, execution time and computational cost of the ML algorithms.

- Per data type, explore with the parameters to determine the optimal value range of parameters per ML algorithm to increase the performance of the developed model.
- Utilise dimensionality reduction techniques in order to improve the performance of the ML algorithms and to compare models with dimensionality reduction to models without dimensionality reduction.
- Experiment with balanced and unbalanced datasets for clustering and classification algorithms in order to determine the effect on the performance of the ML algorithms and to measure the robustness of the different ML algorithms.
- Investigate the influence of different input formats for the ML algorithms given the type of data. For example, compare two-dimensional pixels arrays with one-dimensional pixels vectors for image data.
- Determine the applicability of the framework in different programming languages, including C++, Julia and R.
- Evaluate the framework in different business environments, for example, manufacturing, healthcare and maintenance.
- Implement the framework as an online tool.

## 7.3 Appraisal of research work

After conducting the research study regarding the development of a decision support framework, the researcher established the principles of ML and the process of applying ML to data. The researcher feels confident in having achieved the research objectives of creating a decision support framework, which is applicable to various data types and application purposes, whilst indicating the project management triangle trade-offs. The developed decision support framework is a good guideline to provide a starting point for a semi-skilled analyst given their application purpose and data characteristics. However, the researcher is aware that the datasets in this research were unbalanced and more datasets could be employed. The researcher is also aware of the importance of parameter tuning in developing a ML algorithm.

Utilising the decision support framework requires the user to have an understanding of the application purposes incorporated in the framework, as well as the required data preprocessing before applying the ML algorithms to the data. However, the researcher feels confident that a semi-skilled analyst will be aware of these requirements, and know how to address them, before applying the ML algorithm suggested by the decision support framework.

The decision support framework has not yet been commercialised and is unique.

The research study applied DA and ML principles throughout this entire research, which included the interaction of various components, including data types, DA, application purposes, ML algorithms and value creation *via* the project management triangle.

## 7.4 Concluding remarks

In this final section of the research document, the researcher wishes to share some reflections. The research study that was conducted and documented introduced the researcher to the domains of DA and ML. It was discovered that these domains are large and ever growing in both width and depth. There is no limit to the application domains or industries in which DA an ML can be applied. DA and ML are constantly explored and expanded to accommodate the ever growing variety in data and data structures.

The use of DA and ML creates more possibilities and opportunities to learn from and to use to improve our environment, from improving business practices to identifying cancerous cells. By expanding their DA and ML capabilities, businesses can improve their value creation processes, gain competitive advantage and provide optimised product and service delivery.

# References

Abrahams, A., Fan, W., Wang, G. & Zhang, Z. (2015). An Integrated Text Analytic Framework for Product Defect Discovery. *Production and Operations Management*, **24**, 975–990. 12

Ali, J., Khan, R., Ahmad, N. & Maqsood, I. (2012). Random Forests and Decision Trees. *International Journal of Computer Science Issues(IJCSI)*, **9**, 272–278. 89

Alkhatib, K., Hassan, N., Hmeidi, I. & K Ali Shatnawi, M. (2013). Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm. *International Journal of Business, Humanities and Technology*, **3**, 32–44. 118

Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, Cambridge, Massachusetts, London, England, 2nd edn., ISBN 9780262012430, https://books.google.de/books?id=TtrxCwAAQBAJ&printsec=frontcover&dq=alpaydin+Introduction+to+Machine+Learning+2nd+edition&hl=en&sa=X&ved=0ahUKEwip9bP_3PTlAhUb6KYKHQGjBwMQ6AEIQDAD#v=onepage&q=alpaydin%20Introduction%20to%20Machine%20Learning%202nd%20edition&f=false. 43, 66, 71, 72, 73, 75, 76, 78, 79, 80, 82, 83, 89, 90, 91, 92, 93, 94, 98, 99, 100, 101, 104, 105, 106, 109, 114, 115, 116, 117, 118, 119, 121

Amari, S.I. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, **10**, 251–276. 100

Ansoff matrix image (2018). https://www.executestrategy.net/blog/the-ansoff-matrix-helps-organizations-grow. Accessed: 2018-10-11. xiii, 15

Attoh-Okine, N.O. (1999). Analysis of learning rate and momentum term in back-propagation neural network algorithm trained to predict pavement performance. *Advances in engineering software*, **30**, 291–302. 99, 100

Azevedo, A. & Santos, M.F. (2008). KDD,SEMMA and CRISP-DM: a parallel overview. In *IADIS: European Conference on Data Mining*, vol. 8, 182–185. 44, 45, 47, 48, 50, 51, 52, 53, 54

Babu, R. & Ramakrishnan, K. (2004). Recognition of human actions using motion history information extracted from the compressed video. *Image and Vision Computing*, **22**, 597–607. 90, 91

Babuta, A., Oswald, M., Rinik, C., (RUSI), R. & University of Winchester (2018). Machine Learning Algorithms and Police Decision-Making: Legal, Ethical and Regulatory Challenges. *Whitehall Report 3-18*, 1–45. 13

Bacciu, D., Barsocchi, P., Chessa, S., Gallicchio, C. & Micheli, A. (2014). An experimental characterization of reservoir computing in ambient assisted living applications. *Neural Computing and Applications*, **24**, 1451–1464. 226

Backlund, H., Hedblom, A. & Neijman, N. (2011). A Density-Based Spatial Clustering of Application with Noise. 79

Balanced scorecard image (2018). http://www.jigsawbusinesscoach.com/measures/. Accessed: 2018-10-11. xiii, 15

Balanced Scorecard Institute (2019). Balanced Scorecard Basics. https://www.balancedscorecard.org/BSC-Basics/About-the-Balanced-Scorecard, accessed: 2019-02-18. 14

Balcan, M. (2008). Thesis: New Theoretical Frameworks for Machine Learning. 12

Balcik, B. & Ak, D. (2014). Supplier Selection for Framework Agreements in Humanitarian Relief. *Production and Operations Management*, **23**, 1028–1041. 12

Bandyopadhyay, S. & Paul, T. (2013). Segmentation of Brain Tumour from MRI image – Analysis of K-means and DBSCAN Clustering. *International Journal of Research in Engineering and Science (IJRES)*, **1**, 48–57. 79

Barreiro, J., Labarga, J., Vizan, A. & Rios, J. (2003). Functional model for the development of an inspection integration framework. *International Journal of machine Tools & Manufacture*, **43**, 1621–1632. 12

Basak, D., Pal, S. & Patranabis, D. (2007). Support vector regression. *Neural Information Processing*, **11**, 439–445. 119

Batra, D., Chen, T. & Sukthankar, R. (2008). Space-Time Shapelets for Action Recognition. In *IEEE Workshop on Motion and video Computing*, 1–6. 79, 90

Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, **2**, 1–127. 101

Bewley, A., Guizilini, V., Ramos, F. & Upcroft, B. (2014). Online self-supervised multi-instance segmentation of dynamic objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1296–1303. 79

Bijural, L. (2013). Clustering analysis and its applications. In *Proceedings of National Conference on New Horizons in IT - NCNHITT*, vol. 1, 169–172. xv, 75, 76

Bohanec, M. & Zupan, B. (1997). (marko.bohanec @ ijs.si) and (blaz.zupan @'ijs.si). 224

Boschetto, A., Giordano, V. & Veniali, F. (2013). Surface roughness prediction in fused deposition modelling by neural networks. *International Journal of Advanced Manufacturing Technology*, **67**, 2727–2742. 100, 101, 114

Bose, I. & Mahapatra, R.K. (2001). Business data mining – a machine learning perspective. *Information & Management*, **39**, 211–225. 41, 76

Bottou, L. (2012). Lecture Notes in Computer Science (LNCS): Stochastic Gradient Descent Tricks. In *Neural Networks, Tricks of the Trade*, vol. 7700, 421–436, Springer, Berlin, Heidelberg. 100

Bramer, M. (2007). *Principles of Data Mining*. Springer, ISBN 9781846287664, https://books.google.co.za/books?id=xVW7NslHNHsC. 56, 57, 58, 61, 62, 73

Brown, M. & Brocklebank, J. (1997). Data Mining. In *22th SAS Users Group International Conference*, 1–4. 43, 52, 53, 54, 68

Bryman, A., Bell, E., , Hirschsohn, P. & Du Toit, J. (2017). *Research Methodology: Business and Management Contexts*. Oxford Universuty Press, 7th edn., ISBN 9780199076130, https://books.google.de/books?id=0y0VrgEACAAJ&dq=Research+Methodology:+Business+and+Management+Contexts&hl=en&sa=X&ved=0ahUKEwiPu_nPxfHlAhVDwAIHHWUXDu8Q6AEIKTAA. 18, 19

Buciumas, S. & Priestly, J. (2016). Combining Logistic Regression and Time Series Analysis on Commercial Data for modeling Credit and Default Risk. https://www.sas.com/content/dam/SAS/en_us/doc/event/analytics-experience-2016/combining-logistic-regression-time-series-analysis-commercial-data-modeling-credi pdf, accessed: 2019-04-26. 90

Burns, P. (2016). *Entrepreneurship and Small Business: start-up, growth and maturity*. PALGRAVE MACMILLAN, UK, 4th edn., ISBN 9781137430359, https://books.google.de/books?id=8JlMDwAAQBAJ&printsec=frontcover&dq=4th+edition+Entrepreneurship+and+Small+Business:+start-up,+growth+and+maturity&hl=en&sa=X&ved=0ahUKEwjKwcrJ0fHlAhVB26QKHTIiAeIQ6AEIMzAB#v=onepage&q=4th%20edition%20Entrepreneurship%20and%20Small%20Business%3A%20start-up%2C%20growth%20and%20maturity&f=false. 9

Candanedo, L. & Feldheim, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings*, **112**, 28–39. 226

Candanedo, L., Feldheim, V. & Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, **140**, 81–97. 231

Carcano, E.C., Bartolini, P., Muselli, M. & Piroddi, L. (2008). Jordan recurrent neural network versus IHACRES in modelling daily streamflows. *Journal of Hydrology*, **362**, 291–307. 101

Carreira-Perpiñán, M.Á. (2015). A review of mean-shift algorithms for clustering. *ArXiv*, **abs/1503.00687**, 1–28. 79, 85

Casale, P., Pujol, O. & Radeva, P. (2012). Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing - PUC*, **16**, 1–18. 226

Chang, C.c. & Lin, C.j. (2011). LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2**, 1–39. 91, 107, 119

Chaovalitwongse, W.A., Fan, Y. & Sachdeo, R.C. (2007). On the Time Series $K$-Nearest Neighbor Classification of Abnormal Brain Activity. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, **37**, 1005–1016. 90

Checkland, P. (1981). *Systems Thinking, Systems Practice*. John Wiley & Sons, Chichester, 10th edn., ISBN 9780471279112, https://books.google.de/books?id=icXaAAAAMAAJ&q=Systems+Thinking,+Systems+Practice+checkland+0+471+27911+0&dq=Systems+Thinking,+Systems+Practice+checkland+0+471+27911+0&hl=en&sa=X&ved=0ahUKEwjd9Y3ZyfHlAhWBjqQKHZ3BD-QQ6AEINjAC. 19, 20

Checkland, P. (2000). Soft Systems Methodology: A Thirty Year Retrospective. *Systems Research and Behavioral Science*, **17**, S11–S58. 19, 21

Checkland, P. & Poulter, J. (2006). *Learning For Action: A Short Definitive Account of Soft Systems Methodology, and its use for Practitioners, Teachers and Students*. John Wiley & Sons, Chichester, ISBN 780470025543, https://books.google.de/books?id=4pUoAQAAMAAJ&q=Learning+For+Action:+A+Short+Definitive+Account+of+Soft+Systems+Methodology,+and+its+use+for+Practitioners,+Teachers+and+Students&dq=Learning+For+Action:+A+Short+Definitive+Account+of+Soft+Systems+Methodology,+and+its+use+for+Practitioners,+Teachers+and+Students&hl=en&sa=X&ved=0ahUKEwic26H5yfHlAhVCzaQKHUgODQwQ6AEIKTAA. xiii, 19, 20, 21

Chen, C., Gong, Y., De Koster, R. & Van Nunen, J. (2010). A Flexible Evaluative Framework for Order Picking Systems. *Production and Operations Management*, **19**, 70–82. 12

Chen, Q., Luley, R.S., Wu, Q., Bishop, M., Linderman, R.W. & Qiu, Q. (2018). AnRAD: A Neuromorphic Anomaly Detection Framework for Massive Concurrent Data Streams. *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 1622–1636. 13

Cheng, Q., Varshney, P.K. & Arora, M.K. (2006). Logistic Regression for Feature Selection and Soft Classification of Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, **3**, 491–494. 90

Choi, S.H., Jeong, Y. & Jeong, M.K. (2010). A Hybrid Recommendation Method with Reduced Data for Large-Scale Application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **40**, 557–566. 90

Cimiano, P., Hotho, A. & Staab, S. (2004). Comparing Conceptual, Divisive and Agglomerative Clustering for Learning Taxonomies from Text . In *Proceedings of the 16th European Conference on Artificial Intelligence*, 435–439. 79

Clark, A. (2018). The Machine Learning Audit—CRISP-DM Framework. *Information Systems Audit and Control Association (ISACA) Journal*, **1**, 1–6. 44, 46, 47, 48, 49, 50, 63

Comaniciu, D., Ramesh, V. & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR (Cat. No.PR00662)*, vol. 2, 142–149. 79

Conde, A., Arriandiaga, A., Sanchez, J.A., Portillo, E., Plaza, S. & Cabanes, I. (2018). High-accuracy wire electrical discharge machining using artificial neural networks and optimization techniques. *Robotics and Computer-Integrated Manufacturing*, **49**, 24–38. 101, 113

Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D. & Figari, M. (2014). Machine Learning Approaches for Improving Condition Based Maintenance of Naval Propulsion Plants. *Journal of Engineering for the Maritime Environment*. 229

Corne, R., Nath, C., Mansori, M.E. & Kurfess, T. (2016). Enhancing Spindle Power Data Application with Neural Network for Real-time Tool Wear/Breakage Prediction During Inconel Drilling. *Procedia Manufacturing*, **5**, 1–14. 100, 113

Corrigan, D., deRoos, D., Deutsch, T., Parasuraman, K., Zikopoulos, P. & Giles, J. (2012). *Harness the Power of Big Data: The IBM Big Data Platform*. McGraw-Hill Osborne Media, ISBN 9780071808170, https://mhebooklibrary.com/doi/book/10.1036/9780071808187Book DOI. 1, 3

Cortes, C. & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, **20**, 273–297. 91, 103

Cortez, P. & Morais, A. (2007). A Data Mining Approach to Predict Forest Fires using Meteorological Data. In *Proceedings of the 13th EPIA - Portuguese Conference on Artificial Intelligence*, 512–523. 229

Cortez, P. & Silva, A. (2008). Using Data Mining to Predict Secondary School Student Performance. In *Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC)*, 5–12. 228

Cortez, P., Cerdeira, A., Almeida, F., Matos, T. & Reis, J. (2014). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, **47**, 547–553. 229

Coussement, K. & den Poel, D.V. (2009). Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Systems with Applications*, **36**, 6127 – 6134. 89

Creswell, J.W. (2003). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*. SAGE publications, 2nd edn., iSBN 0761924418(c), https://books.google.de/books?id=nSVxmN2KWeYC&dq=Research+Design:+Qualitative,+Quantitative+and+Mixed+Methods+Approaches&hl=en&sa=X&ved=0ahUKEwjQid6___DlAhWG-aQKHX6VATQQ6AEINzAC. 18, 19

Criminisi, A., Shotton, J. & Konukoglu, E. (2012). Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends® in Computer Graphics and Vision*, **7**, 81–227. 12

Das, G., Lin, K.I., Mannila, H., Renganathan, G. & Smyth, P. (1998). Rule Discovery from Time Series. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD)*, 16–22, AAAI Press. 79

De Vito, S., Massera, E., Piga, M., Martinotto, L. & Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B Chemical*, **129**, 750–757. 230

Desale, D. (2016). Top 15 Frameworks for Machine Learning Experts. https://www.kdnuggets.com/2016/04/top-15-frameworks-machine-learning-experts.html, accessed: 2019-02-13. 13

Deselaers, T., Keysers, D. & Ney, H. (2005). Discriminative training for object recognition using image patches. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 157–162. 91

Du Preez (2019). Professor at the Department of Electrical and Electronic Engineering, Stellenbosch University. Accessed: 2019-10-17. 174, 175, 176, 177, 178, 179, 180

Du Preez, A. & Oosthuizen, G. (2018). Machine learning in additive manufacturing as enabler for smart sustainable manufacturing: a review. In *Additive manufacturing as a key driver of the 4th industrial revolution: Proceedings of the 19th Annual Conference of the Rapid Product Development Association of South Africa (RAPDASA)*, 12–21. 39

Du Preez, A. & Oosthuizen, G. (2019a). Machine Learning in Additive and Subtractive Manufacturing as Enabler for Smart Sustainable Manufacturing: A Review. In *Knowledge valorisation in the age of digitalization: Proceedings of the 7th International Conference on Competitive Manufacturing (COMA)*, 301–309. 39

Du Preez, A. & Oosthuizen, G. (2019b). Machine learning in cutting processes as enabler for smart sustainable manufacturing. *Procedia Manufacturing*, **33**, 810 –817, sustainable Manufacturing for Global Circular Economy: Proceedings of the 16th Global Conference on Sustainable Manufacturing (GCSM). 39

Dua, D. & Graff, C. (2017). UCI Machine Learning Repository. 220

Eads, D., Glocer, K., Perkins, S. & Theiler, J. (2005). Grammar-guided feature extraction for time series classification. *Neural Information Processing Systems (NIPS)*, 1–8. 91

Economist Intelligence Unit (2014). Gut & gigabytes: Capitalising on the art & science in decision makings. https://www.pwc-wissen.de/pwc/en/shop/all_publications/Gut+und+gigabytes+Capitalising+on+the+art/?card=12832&variant=PD&action=shop_add_item_to_basket&action_id=29183, https://www.pwc.com/gx/en/issues/data-and-analytics/big-decisions-survey/assets/big-decisions2014.pdf, accessed: 2019-02-11. 7

Eghbal-zadeh, H., Schedl, M. & Widmer, G. (2015). Timbral modeling for music artist recognition using i-vectors. In *23rd European Signal Processing Conference (EUSIPCO)*, 1286–1290. 91

Şeker, S., Ayaz, E. & Türkcan, E. (2003). Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery. *Engineering Applications of Artificial Intelligence*, **16**, 647–656. 101

El-Maleh, K., Klein, M., Petrucci, G. & Kabal, P. (2000). Speech/music discrimination for multimedia applications. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, vol. 4, 2445–2448. 90

English Oxford Living Dictionaries (2019). Definition of data in English. https://en.oxforddictionaries.com/definition/data, accessed: 2019-02-18. 55

FinancesOnline (2019). 20 Best Data Analytics Software for 2019. https://financesonline.com/data-analytics/, accessed: 2019-02-13. 11

Fink, M. & Kraus, S. (2009). *The Management of Small and Medium Enterprises*. Routledge, ISBN 9780415467247, https://books.google.de/books?id=bz6OAgAAQBAJ&printsec=frontcover&dq=The+Management+of+Small+and+Medium+Enterprises&hl=en&sa=X&ved=0ahUKEwjJqPbYz_HlAhXIjKQKHdX_DG4Q6AEIMTAB#v=onepage&q=The%20Management%20of%20Small%20and%20Medium%20Enterprises&f=false. 9

Forsyth, R. (15 march 1990). 8 Grosvenor Avenue Mapperley Park Nottingham NG3 5DX 0602-621676. 224, 228

Fritchoff, D. (2010). *Encyclopedia of Research Design*, vol. 2nd. SAGE Publications, ISBN 9781412961271, https://books.google.co.za/books?id=pvo1SauGirsC. 61, 62

Gantz, J. & Reinsel, D. (2012). The Digital Universe In 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. In *Proceedings of IDC iView: Analyze the Future*, vol. 2007, 1–16. 1, 2, 4, 5

Garcia-Romero, D. & Espy-Wilson, C.Y. (2010). Automatic acquisition device identification from speech recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1806–1809. 91

Gary Bradshaw, g. (March 1998). 228

Gasson, S. (1994). The Use of Soft Systems Methodology (SSM) As A Tool For Investigation. xiii, 20, 21

Ghorbani, A. & Farzai, S. (2018). Fraud Detection in Automobile Insurance using a Data Mining Based Approach. *international Journal of Mechatronics, Electrical and Computer Technology (IJMEC)*, **8**, 3764–3771. 79

Glasberg, R., Schmiedeke, S., Mocigemba, M. & Sikora, T. (2008). New Real-Time Approaches for Video-Genre-Classification Using High-Level Descriptors and a Set of Classifiers. In *IEEE International Conference on Semantic Computing*, 120–127. 89

Gorban, A., Grechuk, B. & Tyukin, I. (2018). Augumented artificial intelligence: a conceptual framework. *arXiv preprint, arXiv:1802.02172*, **abs/1802.02172**, 1–13. 12

Gorelick, L., Blank, M., Shechtman, E., Irani, M. & Basri, R. (2007). Actions as Space-Time Shapes. *Transactions on Pattern Analysis and Machine Intelligence*, **29**, 2247–2253. 223

Greene, J.C., Caracelli, V.J. & Graham, W.F. (1989). Toward a Conceptual Framework for Mixed-Method Evaluation Designs. *Educational Evaluation and Policy Analysis*, **11**, 255–274. 18

Grobler (2019). Associate professor at the Department of Industrial Engineering, Stellenbosch University. Accessed: 2019-10-14. 174, 175, 176, 178, 179

Guillén, R., Jensen, C. & Edelson, S. (2010). A machine learning approach for identifying subtypes of autism. In *Proceedings of the 1st ACM International Health Informatics Symposium*, 620–628, ACM, New York, NY, USA. 74

Gupta, B., Rawat, A., Jain, A., Arora, A. & Dhami, N. (2017). Analysis of Various Decision Tree Algorithms for Classification in Data Mining. *International Journal of Computer Applications*, **163**, 15–19. 89, 111, 112, 118

Harrag, F., El-Qawasmeh, E. & Pichappan, P. (2009). Improving arabic text categorization using decision trees. In *First International Conference on Networked Digital Technologies*, 110–115. 89

Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Company, Greenwich, CT, USA, ISBN 9781617290183, https://books.google.de/books?id=2d7RXwAACAAJ&dq=harrington+machine+learning+in+action&hl=en&sa=X&ved=0ahUKEwjU14zGmfHlAhWmxMQBHfgNAxIQ6AEIKTAA. 58, 66, 72, 75, 79, 82, 84, 89, 90, 93, 94, 95, 96, 97, 100, 108, 109, 111, 113, 117, 118, 120

Harrison, D. & Rubinfeld, D. (1978). Hedonic prices and the demand for clean air. *J. Environ. Economics & Management*, **5**, 81–102. 229

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edn., ISBN 9780387848587, http://dx.doi.org/10.1007/b94608. 56, 57

Hellerstein, J.M. (2008). Quantitative Data Cleaning for Large Databases. *United Nations Economic Commission for Europe (UNECE)*, 1–42. 59, 60

Helwig, N., Pignanelli, E. & Schütze, A. (2015). Condition monitoring of a complex hydraulic system using multivariate statistics. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 210–215. 227, 231

Herbst (2019). Employee at Prealexis and former professor from the Department of Electrical and Electronic Engineering, Stellenbosch University. Accessed: 2019-10-18. 174, 175, 176, 178, 179, 180

Heysem Kaya, P.T. & Gürgen, S.F. (2012). Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine. In *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE*, 13–18. 229

Hill, A. (2017). Sustaining Growth with the Three Horizons Model for Innovation. https://medium.com/frameplay/planning-for-future-growth-with-the-three-horizons-model-for-innovation-18ab29086 Accessed: 2019-02-18. 14

Hinton, G., Srivastava, N. & Swersky, K. (2012). Lecture 6.5 — Rmsprop: normalize the gradient [Neural Networks for Machine Learning. 100

Hoffmann (2019). CEO and co-founder of Praelexis, Technopark, Stellenbosch. Accessed: 2019-10-18. 174, 175, 176, 177, 178, 179, 180

Holst, A. (2017). Information created globally 20102025. https://www.statista.com/statistics/871513/worldwide-data-created/, Accessed: 2019-08-13. 5

Hung, C. & Tsai, C.F. (2008). Market Segmentation Based on Hierarchical Self-organizing Map for Markets of Multimedia on Demand. *Expert Syst. Appl.*, **34**, 780–787. 79

Iliou, T. & Anagnostopoulos, C.N. (2010). Classification on Speech Emotion Recognition - A Comparative Study. *International Journal on Advances in Life Sciences*, **2**, 18–28. 90

J., G. & C., A.H.B. (2009). Dynamic Image Segmentation Method Using Hierarchical Clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP)*, vol. 5856, 1–7, Springer, Berlin, Heidelberg. 79

Jabareen, Y. (2009). Building a Conceptual Framework: Philosophy, Definitions, and Procedure. *International Journal of Qualitative Methods*, **8**, 49–62. xiii, 27, 28, 30, 31

Jeffrey C. Schlimmer, J..a. (May 1987). 228

Jiménez, F., Sánchez, G., García, J.M., Sciavicco, G. & Miralles, L. (2017). Multi-objective evolutionary feature selection for online sales forecasting. *Neurocomputing*, **234**, 75–92. 77

Jotheeswaran, J. & Kumaraswamy, Y. (2013). Opinion mining using decision tree based feature selection through Manhattan hierarchical cluster measure. *Journal of Theoretical and Applied Information Technology*, **58**, 72–80. 89

Junejo, I., A Bhutta, A. & Foroosh, H. (2010). Dynamic scene modeling for object detection using single-class SVM. In *Proceeding of International Conference on Image Processing, vol. 1*, 1541–1544. 79

Kahanda, I. & Neville, J. (2009). Using Transactional Information to Predict Link Strength in Online Social Networks. In *Proceedings of the Third International ICWSM Conference*, 74–81. 90

Kampouraki, A., Manis, G. & Nikou, C. (2009). Heartbeat Time Series Classification With Support Vector Machines. *IEEE Transactions on Information Technology in Biomedicine*, **13**, 512–518. 91

Kart, L. & Heudecker, N. (2015). Survey Analysis: Practical Challenges Mount as Big Data Moves to Mainstream. Gartner. http://www.gartner.com/newsroom/id/313o817, accessed: 2019-02-11. 7

Khan, R., Hanbury, A. & Stoettinger, J. (2010). Skin detection: A random forest approach. In *IEEE International Conference on Image Processing*, 4613–4616. 89, 90, 91

Kidd (2019). Director of the Centre of Statistical Consultation (CSC), Stellenbosch University. Accessed: 2019-10-17. 175, 176, 178, 180

Kim, H., Howland, P. & Park, H. (2005). Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research*, **6**, 37–53. 91

Kim, K.I., Jung, K. & Kim, H.J. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, **9**, 40–42. 91

Kingma, D.P. & Ba, J.L. (2015). Adam: A Method for Stochastic Optimization. *International and Comparative Law Review (ICLR)*, **abs/1412.6980**, 1–15. 100

Köksal, G., Batmaz, I. & Testik, M.C. (2011). Review: A Review of Data Mining Applications for Quality Improvement in Manufacturing Industry. *Expert Systems with Applications*, **38**, 13448–13467. 39, 42, 72

Koskela, T., Lehtokangas, M., Saarinen, J. & Kaski, K. (1997). Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. In *Proceedings of the World Congress on Neural Networks*, 1–5. 119

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *International Conference on Computer Vision*, 2556–2563. 223

Kumar, S. & Antonenko, P.D. (2014). Connecting practice, theory & method: Supporting professional doctoral students in developing conceptual frameworks. *TechTrends: Linking Research & Practice to Improve Learning*, **58**, 54–61. 27

Lamrini, B., Gjini, A., Daudin, S., Armando, F., Pratmarty, P. & Travé-Massuyès, L. (2018). Anomaly Detection Using Similarity-based One-Class SVM for Network Traffic Characterization. In *29th International Workshop on Principles of Diagnosis*, vol. 2289, 1–8. 79

Laurier, C., Grivolla, J. & Herrera, P. (2008). Multimodal Music Mood Classification Using Audio and Lyrics. In *Seventh International Conference on Machine Learning and Applications*, 688–693. 90, 91

Lawrence, R. & Wrlght, A. (2001). Rule-Based Classification Systems Using Classification and Regression Tree (CART) Analysis. *Photogrammetric Engineering and Remote Sensing*, **67**, 1137 – 1142. 89

Lee, H., Kim, S.G., Park, H.W. & Kang, P. (2014). Pre-launch new product demand forecasting using the Bass model: A statistical and machine learning-based approach. *Technological Forecasting and Social Change*, **86**, 49–64. 77

LI, Y. & Jain, A. (1998). Classification of Text Documents. *The Computer Journal*, **41**, 537–546. 91

Liaw, A. & Wiener, M. (2001). Classification and Regression by RandomForest. *R News: The Newsletter of the R Project*, **2**, 18–22. 118

Lin, B., Huangfu, Y., Lima, N., Jobson, B., Kirk, M., O'Keeffe, P., Pressley, S., Walden, V., Lamb, B. & Cook, D. (2017). Analyzing the Relationship between Human Behavior and Indoor Air Quality. *Journal of Sensor and Actuator Networks*, **6**, 1–18. 77

Lin, Y., Jiang, J. & Lee, S. (2014). A Similarity Measure for Text Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, **26**, 1575–1590. 79, 90

Lu, L., Zhang, H.J. & Li, S. (2003). Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, **8**, 482–492. 91

Lyu, S. & Farid, H. (2003). Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines. In *Revised Papers from the 5th International Workshop on Information Hiding (IH)*, 340–354, Springer-Verlag, London, UK, UK. 91

Ma, J. & Perkins, S. (2003). Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, 1741–1745. 79

Manning, C.D., Raghavan, P. & Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, ISBN 9780521865715, https://books.google.de/books?id=t1PoSh4uwVcC&printsec=frontcover&dq=An+Introduction+to+Information+Retrieval&hl=en&sa=X&ved=0ahUKEwjs-r2ej_HlAhVH4qQKHR3OAQ8Q6AEIKTAA#v=onepage&q=An%20Introduction%20to%20Information%20Retrieval&f=false. 71, 74, 75, 79, 82, 84, 104, 106

Mariscal, G., Marbán, O. & Fernández, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. *Knowledge Eng. Review*, **25**, 137–166. xiii, 44, 52, 53, 54

Marsland, S. (2014). *Machine Learning: An Algorithmic Perspective, Second Edition*. CRC Press, Taylor & Francis Group, Chapmann & Hall, 2nd edn., ISBN 9781466583337, https://books.google.de/books?id=y_oYCwAAQBAJ&printsec=frontcover&dq=Machine+Learning:+An+Algorithmic+Perspective,+Second+Edition&hl=en&sa=X&ved=0ahUKEwjsgIXrk_HlAhVrMewKHdJfBS8Q6AEIKTAA#v=onepage&q=Machine%20Learning%3A%20An%20Algorithmic%20Perspective%2C%20Second%20Edition&f=false. 40, 49, 62, 63, 66, 71, 72, 75, 77, 79, 82, 84, 89, 90, 91, 92, 93, 95, 97, 98, 99, 100, 101, 102, 103, 108, 109, 110, 111, 112, 113, 117, 118, 119, 120

Martiniano, A., Ferreira, R.P., Sassi, R.J. & Affonso, C. (2012). Application of a neuro fuzzy network in prediction of absenteeism at work. In *Proceedings of the 7th Iberian Conference, Information Systems and Technologies (CISTI)*, 1–4. 230

Martínez, F., Pilar Frías, M., Dolores Pérez, M. & Rivera Rivas, A. (2017). A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, 1–19. 118

Mass, J., Srirama, S.N., Flores, H. & Chang, C. (2014). Proximal and Social-aware Device-to-device Communication via Audio Detection on Cloud. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, MUM '14, 143–150, ACM, New York, NY, USA. 79

Maynard, R. (2017). Scope REALLY Matters. https://maynardonline.net/2017/02/26/scope-really-matters/, Accessed: 2019-02-13. xiii, 2, 3

McCallum, A. & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, 41–48, AAAI Press. 91

McKinsey's strategic horizons image (2018). https://www.executestrategy.net/blog/mckinseys-three-horizons-of-growth. Accessed: 2018-10-11. xiii, 15

MEITY (2018). Ministry of Electronics and Information Technology (MEIT), Govt.of India. 224

Mirhoseini, A., Dyer, E., Songhori, E., Baraniuk, R., & Koushanfar, F. (2018). RankMap: A Framework for Distributed Learning From Dense Data Sets. *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 2717–2730. 12

Misra, R. & Arora, P. (2019). Sarcasm Detection using Hybrid Neural Network. *arXiv preprint arXiv:1908.07414*. 218

MIT OpenCourseWare (2014). 16. Learning: Support Vector Machines. 91

Moin, K. & Ahmed, Q. (2012). Use of Data Mining in Banking. *International Journal of Engineering Research and Applications (IJERA)*, **2**, 738–742. xv, 75

Montclair State University (2017). Various kinds of data types known in statistics. http://pages:csam:montclair:edu/~mcdougal/SCP/D_types:htm, accessed: 2019-02-18. 57

Moore, D., McGabe, G. & Bruce, C. (2009). *Introduction to the Practice of Statistics*. W H Freeman and Company, New York, USA, 6th edn., ISBN 9781429216227, https://books.google.de/books?id=FiZ7SgAACAAJ&dq=6th+edition+introduction+to+the+practise+of+statistics&hl=en&sa=X&ved=0ahUKEwiUp5-9s_HlAhVx66YKHUFsAUUQ6AEILDAA. 43, 58

Muehlhausen, J. (2012). The difference between the business model, framework and architecture. http://customerthink.com/the_difference_between_the_business_model_framework_and_architecture/, accessed: 2019-02-18. 14

Mujawar, S. & Joshi, A. (2015). Data Analytics Types, Tools and their Comparison. *International Journal of Advanced Research in Computer and Communication Engineering*, **4**, 488–491. xiii, 38, 39

Nanopoulos, A., Alcock, R. & Manolopoulos, Y. (2001). Feature-based Classification of Time-series Data. *International Journal of Computer Research*, **10**, 49–61. 90

Neural netwotk image (2018). http://www.texample.net/tikz/examples/neural-network/. Accessed: 2019-10-11. xiv, 99

Ng, A. & Stanford University (2016). Artificial Intelligence - All in one. 90, 97, 108

Ngai, E., Hu, Y., Wong, Y.H., Chen, Y. & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, **50**, 559–569. 39, 41, 42, 74

Niaksu, O. (2015). CRISP Data Mining Methodology Extension for Medical Domain. *Baltic Journal of Modern Computing*, **3**, 92–109. 46, 48, 50

Niimi, A. (2015). Deep learning for credit card data analysis. In *World Congress on Internet Security (WorldCIS)*, 73–77. 91

Nisbet, R., Elder, J. & Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press, Inc., Orlando, FL, USA, 1st edn., ISBN 978-0123747655, https://books.google.de/books?id=U5np34a5fmQC&printsec=frontcover&dq=Handbook+of+Statistical+Analysis+and+Data+Mining+Applications&hl=en&sa=X&ved=0ahUKEwjanbvgsvHlAhVE2qQKHTQgC-UQ6AEIMzAB#v=onepage&q=Handbook%20of%20Statistical%20Analysis%20and%20Data%20Mining%20Applications&f=false. xiii, xv, 4, 5, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 58, 59, 62, 63, 64, 65, 66, 67, 68, 72, 73, 74, 85

Niu, L., Li, W. & Xu, D. (2018). An Exemplar-Based Multi-View Domain Generalization Framework for Visual. *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 259–272. 13

Oliver, A., Freixenet, J. & Zwiggelaar, R. (2005). Automatic classification of breast density. In *IEEE International Conference on Image Processing*, vol. 2, 1–4. 89

Omid Sadjadi, S., Ahadi, S.M. & Hazrati, O. (2007). Unsupervised speech/music classification using one-class support vector machines. In *6th International Conference on Information, Communications Signal Processing*, 1–5. 79

Oosthuizen (2018). The data gap. Retired associate professor from the Department of Industrial Engineering at Stellenbsoch University. xiii, 4

Ordóñez, F., De Toledo, P. & Sanchis de Miguel, A. (2013). Activity Recognition Using Hybrid Generative/Discriminative Models on Home Enviroments Using Binary Sensors. *Sensors (Basel, Switzerland*, **13**, 5460–5477. 226

Paliwal, M. & A. Kumar, U. (2009). Neural networks and statistical techniques: A review of applications. Expert Systems with Applications. *Expert Systems with Applications*, **36**, 2–17. 90, 119

Paredes, M. (2018). *Data Science and Advanced Analytics: An integrated framework for creating value from data*. Ph.D. thesis, Massachusetts Institute of Technology. 13

Paschek, D., Luminosu, C.T. & Draghici, A. (2017). Automated business process management – in times of digital transformation using machine learning or artificial intelligence. *MATEC Web of Conferences*, **121**, 1–8. 72

Patsadu, O., Nukoolkit, C. & Watanapa, B. (2012). Human gesture recognition using Kinect camera. In *Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, 28–32. 89, 91

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830. xiv, xv, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 89, 90, 91, 94, 95, 96, 97, 98, 100, 102, 103, 106, 107, 110, 111, 114, 115, 118, 119, 120, 121

Pei, X., Chen, C. & Guan, Y. (2017). Joint Sparse Representation and Embedding Propagation Learning : A Framework for Graph-Based Semisupervised Learning. *IEEE Transactions on Neural Networks and Learning Systems*, **28**, 2949–2960. 12

Pellegrini, T., Portelo, J., Trancoso, I., Abad, A. & Bugalho, M. (2009). Hierarchical Clustering Experiments for Application to Audio Event Detection. In *13th International Conference on Speech and Compute (SPECOM)*. 79

Pesch, R., Schmidt, G., Schroeder, W. & Weustermann, I. (2011). Application of CART in ecological landscape mapping: Two case studies. *Ecological Indicators*, **11**, 115 – 122, spatial information and indicators for sustainable management of natural resources. 89

Phan, H., Maass, M., Mazur, R. & Mertins, A. (2015). Random Regression Forests for Acoustic Event Detection and Classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **23**, 20–31. 89

Piczak, K.J. (2015). ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, 1015–1018, ACM Press. 221

Pinto Ferreira, R., Martiniano, A., Ferreira, A., Ferreira, A. & Jose Sassi, R. (2016). Study on Daily Demand Forecasting Orders using Artificial Neural Network. *IEEE Latin America Transactions*, **14**, 1519–1525. 230

Pla, F., Carmona, P.L. & Sotoca, J.M. (2013). One-class classification techniques in image recognition problems. In *12th Workshop on Information Optics (WIO)*, 1–3. 79

Plano Clark, V.L. & Ivankova, N.V. (2016). *Mixed Methods Research: A Guide to the Field*. SAGE Publications, ISBN 9781483306759, https://dx.doi.org/10.4135/9781483398341. 18, 19

Portilla, J. (2017). Python for Data Science and Machine Learning Bootcamp. 72

Prasad, Y. (2016). *Big Data Analytics Made Easy*. Notion Press, ISBN 9781946390721, https://books.google.co.za/books?id=43q2DQAAQBAJ. 42

Pretorius (2019). PhD obtained at the Department of Computer Science, Stellenbosch University. Accessed: 2019-10-14. 174, 175, 176, 177, 178, 179, 180

Rajaraman, V. (2016). Big data analytics. *Resonance*, **21**, 695–716. xiii, 4, 5, 37, 38, 39, 55, 57

Ramasesh, R. & Browning, T. (2014). A conceptual framework for tackling knowable unknown unknowns in project management. *Production and Operations Management*, **32**, 190–204. 12

Ramteke, R. & Khachane, M. (2012). Automatic Medical Image Classification and Abnormality Detection Using KNearest Neighbour. *International Journal of Advanced Computer Research*, **2**, 186–192. 90

Rani, S. & Sikka, G. (2012). Recent Techniques of Clustering of Time Series Data: A Survey. *International Journal of Computer Applications*, **52**, 1–9. 79

Ranjith, R., Athanesious, J.J. & Vaidehi, V. (2015). Anomaly detection using DBSCAN clustering technique for traffic video surveillance. In *Seventh International Conference on Advanced Computing (ICoAC)*, 1–6. 79

Ray, S. & H. Turi, R. (1999). Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation. In *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT)*, vol. 1, 1–7. 79

Redmond, M.A. & Baveja, A. (2002). A Data-Driven Software Tool for Enabling Cooperative Information Sharing Among Police Departments. *European Journal of Operational Research*, **141**, 660–678. 229

Rennie, J.D.M., Shih, L., Teevan, J. & Karger, D.R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML)*, 616–623, AAAI Press. 91

Reyes-Ortiz, J.L., Oneto, L., Samà, A., Parra, X. & Anguita, D. (2016). Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomput.*, **171**, 754–767. 226

Riedmiller, M. & Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *IEEE International Conference on Neural Networks*, vol. 1, 586–591, IEEE. 100

Rocco, T. & Plakhotnik, M. (2009). Literature reviews, conceptual frameworks, and theoretical frameworks: Terms, functions, and distinctions. *Human Resource Development Review*, **8**, 120–130. 27, 28

Rosdi, B., Jaafar, H. & Ramli, D. (2015). Finger vein identification using fuzzy-based k-nearest centroid neighbor classifier. In *AIP Conference Proceedings*, 1–5. 90

Rozenberg, G., Bäck, T. & Kok, J.N. (2012). *Handbook of Natural Computing*. Springer, Berlin, Heidelberg, ISBN 9783540929093, https://books.google.de/books?id=ZwXxuAEACAAJ&dq=handbook+of+Natural+Computing&hl=en&sa=X&ved=0ahUKEwiH5JS_kPHlAhUOPFAKHQWTAJ0Q6AEILjAB. 115

Ryoo, M.S. & Aggarwal, J.K. (2010). UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html. 223

S. Moro, P.C. & Rit, P. (June 2014). A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems. 224

Sadeghkhani, I., Branch, N. & Feuillet, R. (2012). Delta-Bar-Delta and directed random search algorithms to study capacitor banks switching overvoltages. *Serbian Journal of Electrical Engineering*, **9**, 217–229. 100

Šajn, L. & Kukar, M. (2011). Image processing and machine learning for fully automated probabilistic evaluation of medical images. *Computer Methods and Programs in Biomedicine*, **104**, 75–86. 77

Sapankevych, N.I. & Sankar, R. (2009). Time Series Prediction Using Support Vector Machines: A Survey. *IEEE Computational Intelligence Magazine*, **4**, 24–38. 114, 115, 119

Sapp, C. (2017). Preparing and Architecting for Machine Learning. https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/preparing_and_architecting_for_machine_learning.pdfDocument, accessed: 2019-02-20. 58

Sapp, C. & Gartner Inc. (2017). Preparing and Architecting for Machine Learning. *Gartner: Technical Professional Advice*, 1–37. 13

Sathyadevi, G. (2011). Application of CART algorithm in hepatitis disease diagnosis. In *International Conference on Recent Trends in Information Technology (ICRTIT)*, 1283–1287. 89

Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J. & Williamson, R.C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computing*, **13**, 1443–1471. 79, 107

Sebe, N., Lew, M.S., Cohen, I., Garg, A. & Huang, T.S. (2002). Emotion recognition using a Cauchy Naive Bayes classifier. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR): Object recognition supported by user interaction for service robots*, vol. 1, 17–20. 91

SEDA (2016). The Small, Medium and Micro Enterprise Sector of South Africa. http://www.seda.org.za/publications/publications/the%20small,%20medium%20and%20micro%20enterprise%20sector%20of%20south%20africa%20commissioned%20by%20seda.pdf, accessed: 2019-03-02. 8, 9, 10

Sen, J. (2018). Stock Price Prediction Using Machine Learning and Deep Learning Frameworks. In *6th International Conference on Business Analytics and Intelligence (ICBAI)*, 1–9. 118

Shafique, U. & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, **12**, 217–222. 45, 47, 48, 50, 51, 52, 53, 54, 58

Sharma, A., Bhuriya, D. & Singh, U. (2017). Survey of stock market prediction using machine learning approach. In *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology - ICECA*, vol. 2, 506–509, IEEE. 76

Sharma, N., Sharma, P., Irwin, D. & Shenoy, P. (2011). Predicting solar generation from weather forecasts using machine learning. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 528–533. 119

Shields, P.M. (1998). Pragmatism as a Philosophy of Science: A Tool for Public Administration. *Journal of Research in Public Administration*, **4**, 195–225. 29

Shields, P.M. & Tajalli, H. (2006). Intermediate Theory: The Missing Link in Successful Student Scholarship. *Journal of Public Affairs Education*, **12**, 313–334. 29, 30

Shravan Kumar, B. & Vadlamani, R. (2017). *Text Document Classification with PCA and One-Class SVM*, 107–115. Springer Singapore. 79

Singh, S. & Giri, M. (2014). Comparative Study Id3, Cart And C4.5 Decision Tree Algorithm: A Survey. *International Journal of Advanced Information Science and Technology (JIAIST)*, **3**, 47–52. 111

Sinha, V. & Wegener, R. (2013). The Value of Big Data: How Analytics Differentiates Winners, publisher = Bain & Company. Accessed: 2019-02-11. 7

Smola, A.J. & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, **14**, 199–222. 91, 119

Sonosy, O.A., Rady, S., Badr, N.L. & Hashem, M. (2016). Machine Learning Techniques for Mining Location-Based Social Networks for Business Predictions. In *Proceedings of the 10th International Conference on Informatics and Systems - INFOS*, 185–190, ACM, New York, NY, USA. 70

Soomro, K., Zamir, A.R. & Shah, M. (Nov 2012). UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild., CRCV-TR-12-01. https://www.crcv.ucf.edu/data/UCF101.php. 223

Spicer, E., D & Sadler-Smith (2006). Organizational Learning in Smaller Manufacturing Firms. *International Small Business Journal*, **24**, 133–158, accessed: 2019-03-02. 8

Spiehler, V. (1987). DABFT ,Diagnostic Products Corporation,(213) 776-0180 (ext 3014) . 225

Spinler, S. & Kretschmer, A. (2013). A School Feeding Supply Chain Framework : Critical Factors for Sustainable Program Design. *Production and Operations Management*, **23**, 990–1001. 12

Steynberg, R. (2016). A framework for identifying the most likely successful underprivileged tertiary bursary applicants. Stellenbosch University. xiii, 56

Stolfi, D.H., Alba, E. & Yao, X. (2017). Predicting Car Park Occupancy Rates in Smart Cities. In *Second International Conference: Smart-CT*, 107–117. 230

Sun, H., Liu, H.X., Xiao, H., He, R. & Ran, B. (2003). Short Term Traffic Forecasting Using the Local Linear Regression Model. In *Proceedings of the 82nd Annual Meeting on Transportation Research board*, 1–19. 118

Sun, W., Wang, C. & Zhang, C. (2017). Factor analysis and forecasting of CO2 emissions in Hebei, using extreme learning machine based on particle swarm optimization. *Journal of Cleaner Production*, **162**, 1095–1101. 76

Svinicki, M. (2008). *A Guidebook On Conceptual Frameworks For Research In Engineering Education*. University of Texas, ISSN 0341-127, http://personal.cege.umn.edu/~smith/docs/RREE-Research_Frameworks-Svinicki.pdf. 29

Tai, K.S., Socher, R. & Manning, C.D. (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1556–1566, Association for Computational Linguistics. 101

Tamatjita, E.N. & Mahastama, A.W. (2016). Comparison of music genre classification using Nearest Centroid Classifier and k-Nearest Neighbours. In *International Conference on Information Management and Technology (ICIMTech)*, 118–123. 90

Tan, S. (2005). Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, **28**, 667 – 671. 90

Tan, S.C. & San Lau, J.P. (2014). Time series clustering: A superior alternative for market basket analysis. In *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng)*, 241–248, Springer, Singapore. 230

Tas, T. & Gorur, a.k. (2007). Author identification for Turkish texts. *Journal of Arts and Sciences*, **7**, 151–161. 90

Tatsumi, K., Yamashiki, Y., Torres, M.A.C. & Taipe, C.L.R. (2015). Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data. *Computers and Electronics in Agriculture*, **115**, 171 – 179. 89

Taylor, A. & Mariton, J. (2012). Our Favorite Business Strategy Frameworks. https://www.smestrategy.net/blog/business-strategy-frameworks-for-strategic-planning, accessed: 2019-02-18. 16

Thurn, S. & Anderson, C. (2017). What AI is – and isn't. Accessed: 2019-02-11. 7, 43, 70

Tien, J. (2013). Big Data: Unleashing information. *Journal of Systems Science and SystemsEngineering*, **22**, 127–151. xiii, 5, 6, 55

Timofeev, R.V. (2004). Classification and Regression Trees(CART) Theory and Applications. In *Masters Thesis: CASE - Center of Applied Statistics and Economics Humboldt University, Berlin*, 1–40. 118

Tsunoo, E., Ono, N. & Sagayama, S. (2009). Musical Bass-Line Pattern Clustering and Its Application to Audio Genre Classification. In *10th International Society for Music Information Retrieval Conference (ISMIR)*, 219–224. 79

Tüfekc, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 126–140. 229

Value disciplines image (2018). https://www.executestrategy.net/blog/value-disciplines. Accessed: 2018-10-11. xiii, 15

Wan, X., Xiong, C., C., Z. & Wang, X. (2008). A unified framework of error evaluation and adjustment in machining. *International Journal of machine Tools & Manufacture*, **48**, 1198–1210. 12

Wang, J., Shen, X. & Pan, W. (2006). On Transductive Support Vector Machines. *Prediction and Discovery*, **443**, 1–9. 104

Wright, T. (2018). 5 of the Best Strategy Frameworks for your Organization. https://www.executestrategy.net/blog/best-strategy-frameworks/, accessed: 2019-02-18. 14, 15

Wu, Q., Yan, Q., Deng, H. & Wang, J. (2010). A combination of data mining method with decision trees building for Speech/Music discrimination. *Computer Speech & Language*, **24**, 257 –272. 89

Wu, Y., Wu, Z. & Fu, C. (2018). Continuous Arm Gesture Recognition Based on Natural Features and Logistic Regression. *IEEE Sensors journal*, **18**, 1–6. 90

Yua, C. & Jin-Sheng, Y. (2011). Text Clustering Based on Improved DBSCAN Algorithm. *CNKI Journal: Electronic Technology & Information Science - Computer Engineering*, **12**. 79

Zhang, T. & J. Oles, F. (2001). Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval*, **4**, 5–31. 90, 91, 118

Zhu, W., Gupta, M., Kumar, V., Perepa, A., Asthi, A. & Statchuk, C. (2014). *Building Big Data and Analytics Solutions in the Cloud*. Redbooks, USA, ISBN 9780738453996, http://www.redbooks.ibm.com/abstracts/redp5085.html?Open. 1, 2, 3, 4, 5, 7, 10, 37, 42

# Appendix A

# Summary of the datasets

The following tables provide information on the different datasets used in this study. For clustering and classification, the columns are as follows: the dataset name, the size of the dataset, an explanation on what the target is, the number of clusters/classes, the dataset reference and the dataset acknowledgement. The size of the dataset have different interpretations, which is explained in Table A.1.

The text datasets were reduced to avoid memory errors during clustering and classification. The ratio of classes were preserved, however, in some cases the size reduction lead to the elimination of a class(es).

In the original format, each instance in the image dataset is an image and an image is a matrix with dimensions width by height of the image, or in colour, which then results in an image being represented by a three-dimensional matrix width by height by RGB (Reg, Green, Blue) values. Each image is transformed such that the matrix becomes a one-dimensional vector of length (width x height x colour). In some cases, the image dimensions were reduced to reduce the memory requirements.

Table A.1: Abbreviations used in the tables

| | |
|---|---|
| o | original |
| p | preprocessed |
| c | clustering and classification |
| clus | clustering |
| clas | classification |

Generally, audio datasets are collections of audio files of either the same or variable length. In the case of uniform length, it is indicated by the number of files by the duration in seconds of each. In the case of variable length, the total length of the entire dataset is indicated in brackets. Samples were taken from each audio file at a constant rate and nineteen features were extracted per sample. The following features were extracted from the audio data sets: zero crossing rate, short time energy, spectrum flux, spectrum rolloff, spectral centroid, mel-frequency cepstral coefficients, spectral bandwidth and pitch. Each is one-dimensional, except for the mel-frequency cepstral coefficients which are twelve-dimensional, overall resulting in nineteen features.

The original format of the video datasets, in order, is the total number of videos, the total number of images and lastly the approximate dimension of each image. The pre-processed datasets are the total number of images by the size of each image transformed to a one-dimensional vector.

The transactional and time-series dataset tables have an additional column, 'Type', which indicates whether the data consists of only categorical, only numerical or a mixture of both types of data.

The time-series dataset tables have an additional column, 'Lag', which indicates the time measurement per lag unit if it is provided by the source.

Table A.2: The text datasets used for clustering and classification

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 1. 20 Newsgroups | o: 18846 x 1<br>p: 18846 x 152843<br>c: 1715 x 27854 | News Topic | p: 20<br>c: 3 | Python<br>Original source | None |
| 2. SPAM text | o: 5572 x 1<br>p: 5572 x 8748<br>c: 5572 x 8748 | Spam or not spam | 2 | Kaggle | Acknow. |
| 3. Advertisements | o: 97601 x 1<br>p: 97601 x 165007<br>c: 2727 x 16547 | Job or real estate | 2 | Kaggle | None |
| | o: 97601 x 1<br>p: 97601 x 165007<br>c: 2640 x 19085 | Advertisement topic | p: 27<br>c: 25 | | |
| 5. Industries | o: 1520 x 1<br>p: 1362 x 13636<br>c: 1362 x 13636 | Industry type | 148 | Kaggle | None |
| 6. Medical | o: 57280 x 1<br>p: 53933 x 10258<br>c: 10787 x 4532 | Medical service description | 6 | Kaggle | None |
| | o: 57280 x 1<br>p: 53933 x 10258<br>c: 10516 x 4491 | Medical service description detailed | p: 22<br>c: 21 | | |
| 8. Sarcasm | o: 26705x 1<br>p: 26644 x 25327<br>c: 4441 x 10316 | Sarcasm or not | 2 | Kaggle | Acknow. (Misra & Arora, 2019) |

Continued on next page

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 9. Reviews | o: 4889657 x 1<br>p: 4889657 x 9842053<br>c: 3259 x 14673 | Which language | 4 | Kaggle | Acknow.1,<br>Acknow.2 |
| | o: 4889657 x 1<br>p: 4889657 x 9842053<br>clus: 3259 x 14660<br>clas: 2271 x 10501 | Sentence topic | p: 17<br>clus: 16<br>clas: 15 | | |

Table A.3: The image datasets used for clustering and classification

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 1. MNIST Sign Lang. | o: 34627 x (28x28)<br>p: 34109 x 784 | Identify alphabetic letter | 24 | Kaggle | Acknow. |
| 2. Face | o: 124 x (90x90x4)<br>p: 124 x 32400 | Human face or dol/manikin face | 2 | Kaggle | Acknow. |
| | o: 124 x (90x90x4)<br>p: 112 x 32400 | Emotion | 4 | | |
| | o: 124 x (90x90x4)<br>p: 112 x 32400 | Age | 5 | | |
| | o: 124 x (90x90x4)<br>p: 112 x 32400 | Ethnicity | 6 | | |
| | o: 124 x (90x90x4)<br>p: 112 x 32400 | Gender | 2 | | |
| 7. Captcha 2 text | o: 308 x (200x50)<br>p: 806 x 850 | Aplhabetic letter or number (0-9) | 36 | Kaggle | None |
| 8. Fashion MNIST | o: 70000 x (28x28)<br>p: 70000 x 784<br>clus: 35000 x 784 | Type of clothing | 10 | Kaggle | Acknow. |
| 9. Digits | o: 5620 x (8x8)<br>p: 5620 x 64 | Number (0-9) | 10 | Source | (Dua & Graff, 2017) |
| 10. lfw people | o:13233 x (47x62)<br>p:1867 x 2914 | People | 19 | Python Original source | None |
| 11. Olivetti faces | o:400 x 4096<br>p:400 x 4096 | People | 40 | Python, Original source | AT&T Laboratories Cambridge |

Table A.4: The audio datasets used for clustering and classification

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 1. Audio Cats and Dogs | o: 277 (total: 1921s)<br>p: 76518 x 19<br>clus: 39791 x 19<br>clas: 76518 x 19 | Cat or dog | 2 | Kaggle | AE-Dataset, FreeSound |
| 2. ESC-50 | o: 2000 x 5s<br>p: 398000 x 19 (199 samples per audio file)<br>clus: 39800 x 19<br>clas: 99500 x 19 | Enviromental sound | 5 | Kaggle | Acknow., (Piczak, 2015) |
| | o: 2000 x 5s<br>p: 398000 x 19<br>clus: 39800 x 19<br>clas: 99500 x 19 | Specific environmental sound | 50 | | |
| 4. RAVDESS Emptional Song Audio | o: 1012 (total: 4705s, approx 5s per file)<br>p: 186657 x 19<br>clus: 36239 x 19<br>clas: 97726 x 19 | Song emotion | 6 | Kaggle | Acknow. |
| 5. GTZAN genre | o: 1000 x 30s<br>p: 1199405 x 19<br>clus: 36076 x 19<br>clas: 96120 x 19 | Music genre | 10 | Source | |
| 6. GTZAN music/speech | o: 128 x 30s<br>p: 153472 x 19<br>clus: 35970 x 19<br>clas: 98318 x 19 | Music or speech | 2 | Source | |

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 7. RAVDESS Emotional Speech Audio | o: 1440 (total: 5328s, approx 4s per file)<br>p: 210276 x 19<br>clus: 39343 x 19<br>clas: 99413 x 19 | Speech emotion | 8 | Kaggle | Acknow. |
| 8. Spoken language | o: 3060 x 10s<br>p: 1219743 x 19<br>clus: 38304 x 19<br>clas: 95760 x 19 | Language spoken | 3 | Kaggle | None |
| | o: 3060 x 10s<br>p: 1219743 x 19<br>clus: 38304 x 19<br>clas: 95760 x 19 | Female or male speaking | 2 | | |
| 10. Researcher's | o: 960 (total: 12000s)<br>p: 383040 x 19<br>clus: 38304 x 19<br>clas: 95760 x 19 | Sound type | 4 | see datasets: ESC-50, GTZAN genre and spoken language | see datasets: ESC-50, GTZAN genre and spoken language |

Table A.5: The video datasets used for clustering and classification

| Dataset | Size | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|
| 1. CK+ | o: 313, 5704, (640 x 490)<br>p: 3018 x 27000 | Which human emotion is performed | 7 | AAG | description |
| 2. Weizmann actions | o: 77, 778, (180 x 144 x 3)<br>p: 686 x 18900 | Which human action is performed | 10 | Weizmann | (Gorelick *et al.*, 2007) |
| | o: 14,158, (180 x 144 x 3)<br>p: 141 x 18900 | Walking os standing | 2 | | |
| | o: 10,89, (180 x 144 x 3)<br>p: 79 x 18900 | Walking or running | 2 | | |
| 5. Human interactions | o: 50, 433, (720 x 480 x 3)<br>p: 383 x 120000 | Which human-human interaction | 6 | SDHA 2010 | (Ryoo & Aggarwal, 2010) |
| | o: 17,154,(720 x 480 x 3)<br>p: 137 x 120000 | hugging or pushing | 2 | | |
| 7. Cartoons by the Researcher | o: 40, 280, (diff size x 3)<br>p: 280 x 230400 | Cartoon or human snippet | 2 | Cartoons, Human | (Kuehne *et al.*, 2011) |
| 8. KTH human actions | o: 100, 63, (120 x 160 x 3)<br>p: 3198 x 19200 | Human action performed | 6 | KTH | KTH Royal Institute of Technology |
| 9. UCF101 | o: 50,869, (320 x 240 x 3)<br>p: 686 x 19200 | Hair blow-dry or taichi | 2 | UCF101 | (Soomro *et al.*, Nov 2012) |
| | o: 125,2209,(320 x 240 x 3)<br>p: 1738 x 19200 | Which human action is performed | 5 | | |

Table A.6: The transactional datasets used for clustering and classification

| Dataset | Size | Type | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|---|
| 1. Audiology (Standardized) Data Set | o: 226 x 69<br>p: 216 x 86 | Cat | Audiology diagnosis | 24 | UCI | UCI |
| 2. Car Evaluation Data Set | o:1728 x 6<br>p: 1728 x 21 | Cat | Car acceptability | 4 | UCI | (Bohanec & Zupan, 1997) , UCI |
| 3. Mushroom classification | o: 8124 x 22<br>p: 8124 x 111 | Cat | Poisonous or not | 2 | Kaggle | None |
| 4. Zoo Animal Classification | o: 101 x 16<br>p: 101 x 21 | Cat | Animal type | 7 | Kaggle | (Forsyth, 15 march 1990) |
| 5. Lenses Data Set | o: 24 x 4<br>p: 24 x 6 | Cat | Contact lense advice | 3 | UCI | UCI |
| 6. German credit risk | o: 1000 x 20<br>p: 1000 x 59 | Mix | Good or bad credit risk client | 2 | UCI | UCI |
| 7. Default of credit card clients dataset | o: 30000 x 23<br>p: 30000 x 32 | Mix | Client defaults or not | 2 | UCI | UCI |
| 8. Employee attrition | o:1470 x 34<br>p: 1470 x 76 | Mix | Empoyee attrition occurs or not | 2 | Kaggle | None |
| 9. Income classification | o: 32561 x 14<br>p: 30162 x 103 | Mix | Income bracket | 2 | Kaggle | None |
| 10. Bank marketing dataset | o: 41189 x 20<br>p: 30489 x 53 | Mix | Subscription or not | 2 | UCI | (S. Moro & Rit, June 2014), UCI |
| 11. Audit dataset | o: 776 x 26<br>p: 772 x 25 | Num | Suspicious firm or not | 2 | UCI | (MEITY , 2018), UCI |

Continued on next page

| Dataset | Size | Type | Goal | # | Reference | Acknow. |
|---|---|---|---|---|---|---|
| 12. Banknote authentication dataset | o: 1372 x 4<br>p: 1372 x 4 | Num | Authentic banknote or not | 2 | UCI | UCI |
| 13. Glass dataset | o: 214 x 9<br>p: 214 x 9 | Num | Type of glass | 6 | Kaggle,UCI | (Spiehler, 1987), UCI |
| 14. APS failure at Scania Trucks dataset | o: 76000 x 170<br>p/clas: 68600 x 141<br>clus: 30000 x 141 | Num | Failures related to APS or not | 2 | UCI | UCI |

Table A.7: The time series datasets used for clustering and classification

| Dataset | Size | Type | Goal | # | Lag | Reference | Acknow. |
|---|---|---|---|---|---|---|---|
| 1. ADLs Recognition Using* | o: 2334 x 7<br>p: 492 x 25 | Cat | The activity performed by user | 10 | 1 lag unit | UCI | (Ordóñez *et al.*, 2013), UCI |
| Binary Sensors Data Set | o: 2334 x 7<br>p: 492 x 27 | Mix | | | | | |
| 3. Financial Distress Prediction | o: 3673 x 86<br>p: 2972 x 169 | Num | Financial distress or not | 2 | 1 lag unit | Kaggle | None |
| 4. Activity Recognition | o: 1923177 x 5<br>p: 1922907 x 7<br>clus: 38065 x 7<br>clas: 100437 x 7 | Num | The activity performed by the user | 7 | 1 lag unit | Kaggle | (Casale *et al.*, 2012) |
| 5. Occupancy Detection Data Set | o: 9752 x 8<br>clus: 9751 x 11<br>clas: 10806 x 11 | Num | Room occupied or not | 2 | 1 minute (1 lag) | UCI | (Candanedo & Feldheim, 2016), UCI |
| 6. SPRHA dataset | o: 10929 x 563<br>p: 9630 x 1123 | Num | The activity performed by the volunteer | 6 | 1 lag unit | UCI | (Reyes-Ortiz *et al.*, 2016), UCI |
| 7. EEG Eye State Data Set | o: 14980 x 15<br>p: 14979 x 29 | Num | .... | 2 | 1 lag unit | UCI | UCI |
| 8. Movement prediction | o: 13197 x 6<br>p: 12883 x 10 | Num | .... | 2 | 1 lag unit | UCI | (Bacciu *et al.*, 2014), UCI |

Continued on next page

| Dataset | Size | Type | Goal | # | Lag | Reference | Acknow. |
|---|---|---|---|---|---|---|---|
| 9. Condition monitoring of hydraulic systems Data Set* | o: 2205 x 43680<br>p: 130095 x 35<br>clus: 40000 x 35<br>clas: 100005 x 35 | Num | The cooler efficiency condition | 3 | 1 minute (1 lag) | UCI | (Helwig *et al.*, 2015), UCI |
| 10. DJIA 30 stock time series* | o: 93612 x 7<br>clus: 40021 x 12<br>clas: 93434 x 12 | Num | The daily closing stock priceincreased or decreased | 2 | 1 day (1 lag) | Kaggle | Kaggle, Github and the Market |

Table A.8: The transactional datasets used for regression

| Dataset | Size | Type | Goal | Reference | Acknow. |
|---|---|---|---|---|---|
| 1. Solar Flare dataset | o: 1066 x 13<br>p: 1066 x 26 | Cat | Rise time of a servomechanism | UCI | UCI |
| 2. Servo Data Set | o: 167 x 4<br>p: 167 x 19 | Cat | Number of C-class solar flares by the region | UCI | (Gary Bradshaw, March 1998), UCI |
| 3. Audiology (Standardized) Data Set | o: 226 x 69<br>p: 216 x 86 | Cat | Audiology diagnosis | UCI | UCI |
| 4. Zoo Animal Classification | o: 101 x 16<br>p: 101 x 21 | Cat | Animal type | Kaggle | (Forsyth, 15 march 1990) |
| 5. Servo Data Set | o: 167 x 4<br>p: 167 x 12 | Mix | Number of C-class solar flares by the region | UCI | (Gary Bradshaw, March 1998), UCI |
| 6. Automobile dataset | o: 205 x 25<br>p: 193 x 65 | Mix | Safety risk of the car | UCI | (Jeffrey C. Schlimmer, May 1987), UCI |
| 7. Auto MPG dataset | o: 398 x 8<br>p: 392 x 25 | Mix | Fuel consumption of the car in miles per gallon | UCI | UCI |
| 8. Student Performance Data Set | o: 649 x 32<br>p: 649 x 45 | Mix | Student performance in Portuguese language | UCI | (Cortez & Silva, 2008), UCI |

Continued on next page

| Dataset | Size | Type | Goal | Reference | Acknow. |
|---|---|---|---|---|---|
| 9. Communities and Crime Data Set | o: 1994 x 127<br>p: 1993 x 100 | Num | Predict total number of violent crimes per 100K population | UCI | (Redmond & Baveja, 2002), UCI |
| 10. Forest Fires Data Set | o: 517 x 12<br>p: 517 x 12 | Num | Predict burned area in forest in hectare | UCI | (Cortez & Morais, 2007), UCI |
| 11. Combined Cycle Power Plant Data Set | o: 9568 x 4<br>p: 9568 x 4 | Num | Predict net hourly electrical energy output in MWatt | UCI | (Tüfekc, 2014), (Heysem Kaya & Gürgen, 2012), UCI |
| 12. Condition Based Maintenance of Naval | o: 11934 x 16<br>p: 11934 x 14 | Num | Predict gas turbine compressor decay state coefficient | UCI | (Coraddu *et al.*, 2014), UCI |
| Propulsion Plants Data Set | o: 11934 x 16<br>p: 11934 x 14 | Num | Predict gas turbine turbine decay state coefficient | | |
| 14. Wine Quality Data Set | o: 4898 x 11<br>p: 4898 x 11 | Num | Predict wine quality | UCI | (Cortez *et al.*, 2014), UCI |
| 15. Boston house price dataset | o: 20640 x 8<br>p: 20640 x 8 | Num | Predict the price of the house | Python | (Harrison & Rubinfeld, 1978) |

Table A.9: The time series datasets used for regression

| Dataset | Size | Type | Goal | Lag | Reference | Acknow. |
|---|---|---|---|---|---|---|
| 1. Bike demand dataset | o: 1461 x 3<br>p: 1460 x 5 | Num | Bike demand in public bike service per day | 1 day | Kaggle | None |
| 2. Sales Transactions Dataset Weekly Data Set | o: 811 x 53<br>p: 40550 x 4 | Num | Product purchases per week | 1 week (1 lags) | UCI | (Tan & San Lau, 2014) |
| 3. Absenteeism at work Dataset Weekly Data Set | o: 740 x 21<br>p: 626 x 23 | Mix | Absenteeism in hours | 1 lag unit | UCI | (Martiniano *et al.*, 2012), UCI, UCI |
| 4. Air Quality Data Set | o: 9357 x 15<br>p: 71 x 23 | Num | hourly averaged NO2 concentration in micrograms per cubic meter | 1 hour (1 lag) | UCI | (De Vito *et al.*, 2008), UCI |
| 5. Individual household electric power consumption dataset | o: 2075259 x 9<br>p: 36343 x 13 | Num | The active energy of an electric water-heater and an air-conditioner, in watt-hour, per minute | 1 minute (1 lag) | UCI | UCI |
| 6. Parking Birmingham dataset | o: 35717 x 4<br>p: 35645 x 4 | Num | The occupancy percentage of the parking area | 1 lag unit | UCI | Birmingham City Council, (Stolfi *et al.*, 2017), UCI |
| 7. Daily Demand Forecasting Orders Data Set | o: 60 x 13<br>p: 59 x 21 | Num | The daily number of total orders | 1 day (1 lag) | UCI | (Pinto Ferreira *et al.*, 2016), UCI |

Continued on next page

| Dataset | Size | Type | Goal | Lag | Reference | Acknow. |
|---|---|---|---|---|---|---|
| 8. Appliances energy prediction Data Set | o: 19735 x 29<br>p: 19734 x 51 | Num | The energy usage in watt-hour | 10 minutes (1 lag) | UCI | (Candanedo *et al.*, 2017), UCI |
| 9. Condition monitoring of hydraulic systems Data Set | o: 2205 x 43680<br>p: 130095 x 36<br>r: 100005 x 35 | Num | The hydraulic accumulator pressure in bar | 1 minute (1 lag) | UCI | (Helwig *et al.*, 2015), UCI |
| 10. DJIA 30 stock time series | o: 93612 x 7<br>p: 93465 x 11 | Num | The daily closing stock price | 1 day (1 lag) | Kaggle | Kaggle, Github and the Market |